最优化方法及其 Matlab 程序设计

马昌凤

2009 年 12 月

内容提要

本书较为系统地介绍了非线性最优化问题的基本理论和算法及其主要算法的 Matlab 程序设计. 主要内容包括 (精确或非精确) 线搜索技术, 最速下降法与 (修正) 牛顿法, 共轭梯度法, 拟牛顿法, 信赖域方法, 非线性最小二乘问题的解法, 约束优化问题的最优性条件, 罚函数法, 可行方向法, 二次规划问题的解法, 序列二次规划法以及附录等. 设计的 Matlab 程序有精确线搜索的 0.616 法和抛物线法, 非精确线搜索的 Armijo 准则, 最速下降法, 牛顿法, 再开始共轭梯度法, BFGS 算法, DFP 算法, Broyden 族方法, 信赖域方法, 求解非线性最小二乘问题的 L-M 算法, 解约束优化问题的乘子法, 求解二次规划的有效集法, SQP 子问题的光滑牛顿法以及求解约束优化问题的 SQP 方法等. 此外, 书中配有丰富的例题和习题, 同时, 作为附录介绍了 Matlab 优化工具箱的使用方法. 本书既注重计算方法的实用性, 又注意保持理论分析的严谨性, 强调数值方法的思想和原理在计算机上的实现.

本书的主要阅读对象是数学与应用数学和信息与计算科学专业的本科生, 应用数学、计算数学和运筹学与控制论专业的研究生, 理工科有关专业的研究生, 对最优化理论与算法感兴趣的教师及科技工作人员. 读者只需具备微积分、线性代数和 Matlab 程序设计方面的初步知识.

前 言

运筹学的理论与方法广泛应用于工业与农业、交通与运输、国防与建筑以及通信与管理等各个部门各个领域; 它主要解决最优计划、最优分配、最优决策以及最佳设计和最佳管理等最优化问题. 本书所介绍的最优化方法又称为数学规划, 是运筹学的一个重要分支, 也是计算数学和应用数学的一个重要组成部分.

本书系统地介绍了非线性优化的理论与方法及其 Matlab 程序设计, 其主要阅读对象是数学与应用数学和信息与计算科学专业的本科生, 应用数学、计算数学和运筹学与控制论专业的研究生, 理工科有关专业的研究生, 对最优化理论与算法感兴趣的教师及科技工作人员. 读者只需具备微积分、线性代数和 Matlab 程序设计方面的初步知识.

本书的主要内容包括: 最优化理论基础; (精确或非精确) 线搜索技术; 最速下降法与 (修正) 牛顿法; 共轭梯度法; 拟牛顿法; 信赖域方法; 非线性最小二乘问题的解法; (约束优化问题的) 最优性条件; 罚函数法; 可行方向法; 二次规划问题的解法; 序列二次规划法以及附录等. 设计的 Matlab 程序有精确线搜索的 0.616 法和抛物线法, 非精确线搜索的 Armijo 准则, 最速下降法, 牛顿法, 再开始共轭梯度法, 对称秩 1 算法, BFGS 算法, DFP 算法, Broyden 族方法, 信赖域方法, 求解非线性最小二乘问题的 L-M 算法, 解约束优化问题的乘子法, 求解二次规划的有效集法, 牛顿-拉格朗日算法, SQP 子问题的光滑牛顿法以及求解约束优化问题的 SQP 方法等. 此外, 书中配有丰富的例题和习题, 同时, 作为附录介绍了 Matlab 优化工具箱的使用方法. 本书既注重计算方法的实用性, 又注意保持理论分析的严谨性, 强调数值方法的思想和原理在计算机上的实现.

本书具有如下特点:

1. 介绍非线性优化中最重要最基础的理论与方法, 它们是研究各种复杂的最优化问题的基础和工具.

2. 最优化方法与 Matlab 程序设计相结合, 采用当前最流行的数学软件 Matlab 编制了主要优化算法的 Matlab 程序. 所有程序都在计算机上经过调试和

运行, 简洁而不乏准确.

3. 本书所给的每一程序之后都给出了相应的计算实例. 这不仅能帮助学生理解程序里所包含的最优化理论知识, 而且对培养学生处理数值最优化问题的能力也大有裨益.

4. 全书每章都配备了一定数量的习题, 习题包括理论分析题和编程实验题, 以加强学生对所学知识的理解和巩固.

本书的编写和出版得到了国家自然科学基金项目 (编号: 10661005) 福建省自然科学基金项目 (编号: 2009J01002) 的部分资助, 在此作者表示由衷的感谢. 作者还要感谢福建师范大学教务处及数学与计算机科学学院给予的帮助和支持. 此外, 本书之所以能够顺利付梓, 在很大程度上要归功于夫人刘菊庄女士给予的理解和支持, 深表感谢.

由于作者水平有限, 加之时间仓促, 书中的缺点和错误在所难免, 敬请专家和读者批评指正.

作者

2009 年 12 月

于福建师范大学

目　录

第一章　最优化理论基础

1.1　最优化问题的数学模型

通俗地说，所谓最优化问题，就是求一个多元函数在某个给定集合上的极值. 几乎所有类型的最优化问题都可以用下面的数学模型来描述:

$$\begin{aligned} &\min f(x), \\ &\text{s.t. } x \in K, \end{aligned} \tag{1.1}$$

这里，K 是某个给定的集合 (称为可行集或可行域)，$f(x)$ 是定义在集合 K 上的实值函数. 此外，在模型 (1.1) 中，x 通常称为决策变量, s.t. 是 subject to (受限于) 的缩写.

人们通常按照可行集的性质对最优化问题 (1.1) 进行一个大致的分类:

- 线性规划和非线性规划. — 可行集是有限维空间中的一个子集;
- 组合优化或网络规划. — 可行集中的元素是有限的;
- 动态规划. — 可行集是一个依赖时间的决策序列;
- 最优控制. — 可行集是无穷维空间中的一个连续子集.

本书主要考虑在工程设计中有着重要应用的非线性规划，其数学模型为

$$\min \quad f(x), \tag{1.2}$$

$$\text{s.t. } h_i(x) = 0,\ i = 1, \cdots, l, \tag{1.3}$$

$$g_i(x) \ge 0,\ i = 1, \cdots, m, \tag{1.4}$$

其中，$f(x)$, $h_i(x)$ $(i = 1, \cdots, l)$ 及 $g_i(x)$ $(i = 1, \cdots, m)$ 都是定义在 $\mathbb{R}^n$ 上连续可微的多元实值函数, 且至少有一个是非线性的. 记

$$E = \{i : h_i(x) = 0\}, \quad I = \{i : g_i(x) \ge 0\}. \tag{1.5}$$

若指标集 $E \cup I = \emptyset$, 称之为无约束优化问题，否则称为约束优化问题. 特别地, 把 $E \neq \emptyset$ 且 $I = \emptyset$ 的优化问题称为等式约束优化问题; 而把 $I \neq \emptyset$ 且 $E = \emptyset$ 的优化问题称为不等式约束优化问题. $f(x)$ 称为目标函数, $h_i(x)$, $g_j(x)$ $(i = 1, \cdots, l; j = 1, \cdots, m)$ 称为约束函数. 此外，通常把目标函数为二次函数而约束函数都是线性函数的优化问题称为二次规划；而目标函数和约束函数都是线性函数的优化问题称为线性规划.

1.2　向量和矩阵范数

在算法的收敛性分析中，需要用到向量和矩阵范数的概念及其有关理论.

设 $\mathbb{R}^n$ 表示实 n 维向量空间，$\mathbb{R}^{n\times n}$ 表示实 n 阶矩阵全体所组成的线性空间. 在这两个空间中，我们分别定义向量和矩阵的范数.

向量 $x \in \mathbb{R}^n$ 的范数 $\|x\|$ 是一个非负数，它必须满足以下条件：

（1） $\|x\| \geq 0$，$\|x\| = 0 \Longleftrightarrow x = 0$;

（2） $\lambda x\| = |\lambda|\|x\|$, $\lambda \in \mathbb{R}$;

（3） $\|x + y\| \leq \|x\| + \|y\|$.

向量 $x = (x_1, \cdots, x_n)^T$ 的 p-范数定义为

$$\|x\|_p = \Big(\sum_{i=1}^n |x_i|^p\Big)^{\frac{1}{p}}. \tag{1.6}$$

常用的向量范数有

1-范数：$\|x\|_1 = \sum\limits_{i=1}^n |x_i|$;

2-范数：$\|x\|_2 = \Big(\sum\limits_{i=1}^n |x_i|^2\Big)^{\frac{1}{2}}$;

∞-范数：$\|x\|_\infty = \max\limits_{1\leq i\leq n} |x_i|$.

矩阵 $A \in \mathbb{R}^{n\times n}$ 的范数是一个非负实数，它除了满足跟向量范数相似的三条性质之外，还必须具备乘法性质：

（4） $\|AB\| \leq \|A\|\|B\|$,　$A, B \in \mathbb{R}^{n\times n}$.

如果一矩阵范数 $\|\cdot\|_\mu$ 相对于某向量范数 $\|\cdot\|$ 满足下面的不等式

（5） $\|Ax\| \leq \|A\|_\mu \|x\|$,　$x \in \mathbb{R}^n$,

则称矩阵范数 $\|\cdot\|_\mu$ 和向量范数 $\|\cdot\|$ 是相容的. 进一步，若存在 $x \neq 0$ 使成立

$$\|A\|_\mu = \max_{x\neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|, \quad A \in \mathbb{R}^{n\times n}, \tag{1.7}$$

则称矩阵范数 $\|\cdot\|_\mu$ 是由向量范数 $\|\cdot\|$ 诱导出来的算子范数，简称算子范数，有时也称为从属于向量范数 $\|\cdot\|$ 的矩阵范数. 此时向量范数和算子范数通常采用相同的符号 $\|\cdot\|$.

不难验证，从属于向量范数 $\|x\|_\infty$, $\|x\|_1$, $\|x\|_2$ 的矩阵范数分别为

$$\|A\|_\infty = \max_{1\le i\le n}\sum_{j=1}^{n}|a_{ij}|,$$

$$\|A\|_1 = \max_{1\le j\le n}\sum_{i=1}^{n}|a_{ij}|,$$

$$\|A\|_2 = \max\{\sqrt{\lambda} \mid \lambda\in\lambda(A^TA)\},$$

它们分别称作行和范数、列和范数和谱范数.

本书在讨论各种迭代算法的收敛性时，通常采用谱范数和按下述方式定义的 F-范数:

$$\|A\|_F = \left(\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}^2\right)^{1/2} = \sqrt{\mathrm{tr}(A^TA)} \tag{1.8}$$

现在我们来讨论向量序列和矩阵序列的收敛性. 我们知道，若 $\{x^{(k)}\}_{k=1}^\infty \subset \mathbb{R}^n$，则

$$\lim_{k\to\infty} x^{(k)} = x \Longleftrightarrow \lim_{k\to\infty} x_i^{(k)} = x_i, \quad i=1,\cdots,n.$$

类似地，若 $\{A^{(k)}\}_{k=1}^\infty \subset R^{n\times n}$，则

$$\lim_{k\to\infty} A^{(k)} = A \Longleftrightarrow \lim_{k\to\infty} a_{ij}^{(k)} = a_{ij}, \quad i,\ j=1,\cdots,n.$$

为了利用范数来描述上述极限，必须建立向量范数的等价定理以及矩阵范数的等价定理.

定理 1 (1) 设 $\|\cdot\|$ 和 $\|\cdot\|'$ 是定义在 R^n 上的两个向量范数，则存在两个正数 c_1, c_2，对所有 $x\in\mathbb{R}^n$ 均成立

$$c_1\|x\| \le \|x\|' \le c_2\|x\|.$$

(2) 设 $\|\cdot\|$ 和 $\|\cdot\|'$ 是定义在 $\mathbb{R}^{n\times n}$ 上的两个矩阵范数，则存在两个正数 m_1, m_2，对所有 $A\in\mathbb{R}^{n\times}$ 均成立

$$m_1\|A\| \le \|A\|' \le m_2\|A\|.$$

下面，我们利用范数的概念来等价地定义向量序列和矩阵序列的收敛性.

定理 2 (1) 设 $\{x^{(k)}\}$ 为 n 维向列序列，$\|\cdot\|$ 为定义在 $\mathbb{R}^n$ 上的向量范数，则

$$\lim_{k\to\infty} x^{(k)} = x \iff \lim_{k\to\infty} \|x^{(k)} - x\| = 0;$$

(2) 设 $\{A^{(k)}\}$ 为 $n\times n$ 矩阵序列，$\|\cdot\|$ 为定义在 $\mathbb{R}^{n\times n}$ 上的向量范数，则

$$\lim_{k\to\infty} A^{(k)} = A \iff \lim_{k\to\infty} \|A^{(k)} - A\| = 0.$$

1.3 函数的可微性与展开

本节主要介绍后文经常需要用到 n 元函数的一阶和二阶导数以及泰勒展开式.

定义 1 设有 n 元实函数 $f(x)$, 其中自变量 $x=(x_1,\cdots,x_n)^T\in\mathbb{R}^n$. 称向量

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \cdots, \frac{\partial f(x)}{\partial x_n}\right)^T \tag{1.9}$$

为 $f(x)$ 在 x 处的一阶导数或梯度. 称矩阵

$$\nabla^2 f(x) = \begin{pmatrix} \dfrac{\partial^2 f(x)}{\partial x_1^2} & \dfrac{\partial^2 f(x)}{\partial x_1\partial x_2} & \cdots & \dfrac{\partial^2 f(x)}{\partial x_1\partial x_n} \\ \dfrac{\partial^2 f(x)}{\partial x_2\partial x_1} & \dfrac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f(x)}{\partial x_2\partial x_n} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial^2 f(x)}{\partial x_n\partial x_1} & \dfrac{\partial^2 f(x)}{\partial x_n\partial x_2} & \cdots & \dfrac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix} \tag{1.10}$$

为 $f(x)$ 在 x 处的二阶导数或 Hesse 矩阵. 若梯度 $\nabla f(x)$ 的每个分量函数在 x 都连续, 则称 f 在 x 一阶连续可微；若 Hesse 阵 $\nabla^2 f(x)$ 的各个分量函数都连续，则称 f 在 x 二阶连续可微.

若 f 在开集 D 的每一点都连续可微，则称 f 在 D 上一阶连续可微；若 f 在开集 D 的每一点都都二阶连续可微，则称 f 在 D 上二阶连续可微.

由上述定义不难发现，若 f 在 x 二阶连续可微，则

$$\frac{\partial^2 f(x)}{\partial x_i\partial x_j} = \frac{\partial^2 f(x)}{\partial x_j\partial x_i}, \quad i,j=1,2,\cdots,n,$$

即 Hesse 阵 $\nabla^2 f(x)$ 是对称阵.

例 1 设二次函数

$$f(x) = c^T x + \frac{1}{2} x^T H x,$$

其中 $c \in \mathbb{R}^n$, $H \in \mathrm{R}^{n\times n}$ 是对称阵. 那么，不难计算它在 x 的梯度及 Hesse 为

$$\nabla f(x) = c + Hx, \quad \nabla^2 f(x) = H.$$

例 2 (泰勒展开). 设函数 $f: \mathbb{R}^n \to \mathbb{R}$ 连续可微，那么

$$\begin{aligned} f(x+h) &= f(x) + \int_0^1 \nabla f(x+\tau h)^T h \mathrm{d}\tau \\ &= f(x) + \nabla f(x+\xi h)^T h, \quad \xi \in (0,1) \\ &= f(x) + \nabla f(x)^T h + o(\|h\|). \end{aligned}$$

进一步, 若函数 f 是二次连续可微的, 则有

$$\begin{aligned} f(x+h) &= f(x) + \nabla f(x)^T h + \int_0^1 (1-\tau) h^T \nabla^2 f(x+\tau h) h \mathrm{d}\tau \\ &= f(x) + \nabla f(x)^T h + \frac{1}{2} h^T \nabla^2 f(x+\xi h) h, \quad \xi \in (0,1) \\ &= f(x) + \nabla f(x)^T h + \frac{1}{2} h^T \nabla^2 f(x) h + o(\|h\|^2) \end{aligned}$$

及

$$\begin{aligned} \nabla f(x+h) &= \nabla f(x) + \int_0^1 \nabla^2 f(x+\tau h)^T h \mathrm{d}\tau \\ &= \nabla f(x) + \nabla^2 f(x+\xi h)^T h, \quad \xi \in (0,1) \\ &= \nabla f(x) + \nabla^2 f(x)^T h + o(\|h\|). \end{aligned}$$

下面简单介绍一下向量值函数的可微性及中值定理. 设有向量值函数 $F = (F_1, F_2, \cdots, F_m)^T : \mathbb{R}^n \to \mathbb{R}^m$. 若每个分量函数 F_i 都是 (连续) 可微的，则称 F 是 (连续) 可微的. 向量值函数 F 在 x 的导数 $F' \in \mathbb{R}^{m\times n}$ 是指它在 x 的 Jacobi 矩阵, 记为 $F'(x)$ 或 $J_F(x)$, 即

$$F'(x) := J_F(x) := \begin{pmatrix} \dfrac{\partial F_1(x)}{\partial x_1} & \dfrac{\partial F_1(x)}{\partial x_2} & \cdots & \dfrac{\partial F_1(x)}{\partial x_n} \\ \dfrac{\partial F_2(x)}{\partial x_1} & \dfrac{\partial F_2(x)}{\partial x_2} & \cdots & \dfrac{\partial F_2(x)}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial F_m(x)}{\partial x_1} & \dfrac{\partial F_m(x)}{\partial x_2} & \cdots & \dfrac{\partial F_m(x)}{\partial x_n} \end{pmatrix}.$$

考虑到标量函数的梯度定义, 有时也把向量函数 F 的 Jacobi 矩阵的转置称为 F 在 x 的梯度，记为

$$\nabla F(x) = J_F(x)^T = \big(\nabla F_1(x), \nabla F_2(x), \cdots, \nabla F_m(x)\big).$$

不难发现，例 2 中关于多元实函数的一些结论可以推广到向量值函数的情形. 例如，若向量值函数 $F:\mathbb{R}^n \to \mathbb{R}^m$ 是连续可微的，则对于任意的 $x,\ h \in \mathbb{R}^n$, 有

$$F(x+h) = F(x) + \int_0^1 \nabla F(x+\tau h)^T h \mathrm{d}\tau.$$

对于向量值函数 F, 也可以定义 Lipschitz 连续性的概念.

定义 2 设向量值函数 $F:\mathbb{R}^n \to \mathbb{R}^m$, $x \in \mathbb{R}^n$, 称 F 在 x 是 Lipschitz 连续的，是指存在常数 $L>0$, 使得对任意的 $y \in \mathbb{R}^n$, 满足

$$\|F(x) - F(y)\| \le L\|x-y\|, \tag{1.11}$$

其中, L 称为 Lipschitz 常数. 若 (1.11) 式对任意的 $x,\ y \in \mathbb{R}^n$ 都成立, 则称 F 在 $\mathbb{R}^n$ 内是 Lipschitz 连续的.

在迭代法的收敛性分析中, 有时需要用到向量值函数的“中值定理”, 现引述如下.

定理 3 设向量值函数 $F:\mathbb{R}^n \to \mathbb{R}^m$ 连续可微, 那么

(1) 对任意的 $x, y \in \mathbb{R}^n$, 有

$$\|F(x) - F(y)\| \le \sup_{0\le t\le 1} \|F'(y+t(x-y))\|\,\|x-y\|.$$

(2) 对任意的 $x, y, z \in \mathbb{R}^n$, 有

$$\|F(y) - F(z) - F'(x)(y-z)\| \le \sup_{0\le t\le 1} \|F'(z+t(y-z)) - F'(x)\|\,\|x-y\|.$$

由上述定理的结论 (2) 可推得下面的结论.

推论 1 设向量值函数 $F:\mathbb{R}^n \to \mathbb{R}^m$ 是连续可微的, 且其 Jacobi 矩阵映射是 Lipschitz 连续的, 即存在常数 $L>0$ 使得

$$\|F'(u) - F'(v)\| \le L\|u-v\|, \quad \forall\, u, v \in \mathbb{R}^n. \tag{1.12}$$

则对任意的 $x, h \in \mathbb{R}^n$, 有

$$\|F(x+h) - F(x) - F'(x)h\| \le \frac{1}{2}L\|h\|^2. \tag{1.13}$$

1.4　凸集与凸函数

本节介绍凸集、锥和多面体的有关概念.

定义 3 设集合 $D \subset \mathbb{R}^n$. 称集合 D 为凸集, 是指对任意的 $x, y \in D$ 及任意的实数 $\lambda \in [0,1]$, 都有 $\lambda x + (1-\lambda)y \in D$.

由上述定义不难知道凸集的几何意义. 即对非空集合 $D \subset \mathbb{R}^n$, 若连接其中任意两点的线段仍属于该集合，则称该集合 D 为凸集.

不难证明凸集的下列基本性质.

引理 1 设 D, D_1, D_2 是凸集，α 是一实数, 那么

(1) $\alpha D = \{y | y = \alpha x,\ x \in D\}$ 是凸集.

(2) 交集 $D_1 \cap D_2$ 是凸集.

(3) 和集 $D_1 + D_2 = \{z | z = x + y,\ x \in D_1,\ y \in D_2\}$ 也是凸集.

例 3 n 维欧氏空间中的 l 个点的凸组合是一个凸集. 即集合

$$\left\{x = \sum_{i=1}^{l} \alpha_i x_i \,\middle|\, x_i \in \mathbb{R}^n,\ \alpha_i \geq 0,\ \sum_{i=1}^{l} \alpha_i = 1\right\}$$

是凸集.

例 4 n 维欧氏空间中的超平面 $H = \{x \big| c^T x = \alpha\}$ 是一个凸集, 其中 $\alpha \in \mathbb{R}$, $c \in \mathbb{R}^n \backslash \{0\}$ 是超平面的法向量. 此外, 下面的四个半空间

(1) 正的闭半空间 $H^+ = \{x | c^T x \geq \alpha\}$,　(2) 负的闭半空间 $H^+ = \{x | c^T x \leq \alpha\}$,

(3) 正的开半空间 $H^+ = \{x | c^T x > \alpha\}$,　(4) 负的开半空间 $H^+ = \{x | c^T x < \alpha\}$.

都是凸集

例 5 以 $x^0 \in \mathbb{R}^n$ 为起点, $d \in \mathbb{R}^n \backslash \{0\}$ 为方向的射线

$$r(x^0; d) := \{x \in \mathbb{R}^n \,|\, x = x^0 + \alpha d,\ \alpha \geq 0\}$$

是凸集.

定义 4 集合 $D \subset \mathbb{R}^n$ 的凸包 (convex hull) 是指所有包含 D 的凸集的交集，记为

$$\mathrm{conv}(D) := \bigcap_{C \supseteq D} C,$$

其中 C 为凸集.

下面我们给出锥和凸锥的定义.

定义 5 设非空集合 $C \subset \mathbb{R}^n$. 若对任意的 $x \in C$ 和任意的实数 $\lambda > 0$，有 $\lambda x \in C$, 则称 C 为一个锥 (cone). 若 C 同时也是凸集, 则称 C 为一个凸锥 (convex cone). 此外, 对于锥 C, 若 $0 \in C$, 则称 C 是一个尖锥 (pointed cone). 相应地, 包含 0 的凸锥称为尖凸锥.

例 6 多面体 $\{x \in \mathbb{R}^n | Ax \geq 0\}$ 是一个尖凸锥, 通常称之为多面锥 (polyhedral cone).

例 7 集合

$$\mathbb{R}^n_+ := \{x \in \mathbb{R}^n | x_i \geq 0,\ i = 1, 2, \cdots, n\}$$

是一个尖凸锥, 通常称之为非负锥 (nonnegative orthant). 相应地, 凸锥

$$\mathbb{R}^n_{++} := \{x \in \mathbb{R}^n | x_i > 0,\ i = 1, 2, \cdots, n\}$$

称为正锥 (positive orthant).

有了凸集的概念之后，就可以定义凸集上的所谓凸函数.

定义 6 设函数 $f : D \subset \mathbb{R}^n \to \mathbb{R}$, 其中 D 为凸集.

(1) 称 f 是 D 上的凸函数, 是指对任意的 $x, y \in D$ 及任意的实数 $\lambda \in [0, 1]$, 都有

$$f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y).$$

(2) 称 f 是 D 上的严格凸函数, 是指对任意的 $x, y \in D$, $x \neq y$ 及任意的实数 $\lambda \in [0, 1]$, 都有

$$f(\lambda x + (1 - \lambda) y) < \lambda f(x) + (1 - \lambda) f(y).$$

(3) 称 f 是 D 上的一致凸函数, 是指存在常数 $\gamma > 0$, 使对任意的 $x, y \in D$ 及任意的实数 $\lambda \in [0, 1]$, 都有

$$f(\lambda x + (1 - \lambda) y) + \frac{1}{2}\lambda(1 - \lambda)\gamma \|x - y\|^2 \leq \lambda f(x) + (1 - \lambda) f(y).$$

凸函数具有下列基本性质.

命题 1 设 f, f_1, f_2 都是凸集 D 上的凸函数, $c_1, c_2 \in \mathbb{R}_+$, $\alpha \in \mathbb{R}$, 则有
(1) $cf_1(x) + c_2 f_2(x)$ 也是 D 上的凸函数.
(2) 水平集
$$\mathcal{L}(f, \alpha) = \{x | x \in D,\ f(x) \le \alpha\}$$
是凸集.

凸集和凸函数在优化理论中起着举足轻重的作用, 但是利用凸函数的定义来判断一个函数是否具有凸性并非一件容易的事情. 如果函数是一阶或二阶连续可微的, 则可利用函数的梯度或 Hesse 阵来判别或验证函数的凸性要相对容易一些. 下面给出几个判别定理.

定理 4 设 f 在凸集 $D \subset \mathbb{R}^n$ 上一阶连续可微，则
(1) f 在 D 上为凸函数的充要条件是
$$f(x) \ge f(x^*) + \nabla f(x^*)^T (x - x^*), \quad \forall\, x^*, x \in D. \tag{1.14}$$
(2) f 在 D 上严格凸的充要条件是，当 $x \ne y$ 时，成立
$$f(x) > f(x^*) + \nabla f(x^*)^T (x - x^*), \quad \forall\, x^*, x \in D. \tag{1.15}$$
(3) f 在 D 上一致凸的充要条件是，存在常数 $c > 0$，使对任意的 $x^*, x \in D$，成立
$$f(x) \ge f(x^*) + \nabla f(x^*)^T (x - x^*) + c\|x -^* \|^2. \tag{1.16}$$

我们知道，在一元函数中，若 $f(x)$ 在区间 (a, b) 上二阶可微且 $f''(x) \ge 0$ (> 0)，则 $f(x)$ 在 (a, b) 内凸（严格凸）. 对于二阶连续可微的多元函数 $f : D \subset \mathbb{R}^n \to \mathbb{R}$, 也可以由其二阶导数 (Hesse 阵) 给出凸性的一个近乎完整的表述.

定义 7 设 n 元实函数 f 在凸集 D 上是二阶连续可微的. 若对一切 $h \in R^n$, 有 $h^T \nabla^2 f(x) h \ge 0$，则称 $\nabla^2 f$ 在点 x 处是半正定的. 若对一切 $0 \ne h \in \mathbb{R}^n$, 有 $h^T \nabla^2 f(x) h > 0$，则称 $\nabla^2 f$ 在点 x 处是正定的. 进一步，若存在常数 $c > 0$, 使得对任意的 $h \in \mathbb{R}^n$, $x \in D$, 有 $h^T \nabla^2 f(x) h \ge c\|h\|^2$，则称 $\nabla^2 f$ 在 D 上是一致正定的.

有了上述定义，我们可以把一元函数关于用二阶导数表述凸性的结果推广到多元函数上.

定理 5 设 n 元实函数 f 在凸集 $D \subset \mathbb{R}^n$ 上二阶连续可微, 则
(1) f 在 D 上是凸的充要条件是 $\nabla^2 f(x)$ 对一切 $x \in D$ 为半正定;
(2) f 在 D 上是严格凸的充分条件是 $\nabla^2 f(x)$ 对一切 $x \in D$ 为正定;
(3) f 在 D 上是一致凸的充要条件是 $\nabla^2 f(x)$ 对一切 $x \in D$ 为一致正定.

注意，$\nabla^2 f$ 正定是 f 严格凸的充分条件而非必要条件.

1.5 无约束问题的最优性条件

本节讨论无约束优化问题

$$\min f(x) \tag{1.17}$$

的最优性条件, 它包含一阶条件和二阶条件. 首先给出极小点的定义, 它分为全局极小点和局部极小点.

定义 8 若对于任意的 $x \in \mathbb{R}^n$, 都有

$$f(x^*) \le f(x),$$

则称 x^* 为 f 的一个全局极小点. 若上述不等式严格成立且 $x \neq x^*$, 则称 x^* 为 f 的一个严格全局极小点.

定义 9 若对于任意的 $x \in N(x^*, \delta) = \{x \in \mathbb{R}^n | \|x - x^*\| < \delta\}$, 都有

$$f(x^*) \le f(x),$$

则称 x^* 为 f 的一个局部极小点, 其中 $\delta > 0$ 为某个常数. 若上述不等式严格成立且 $x \neq x^*$, 则称 x^* 为 f 的一个严格局部极小点.

由上述定义可知, 全局极小点一定是局部极小点, 反之不然. 一般来说, 求全局极小点是相当困难的, 因此通常只求局部极小点 (在实际应用中, 有时求局部极小点已满足了问题的要求). 故本书所讨论的求极小点的方法都是指局部极小点.

为了讨论和叙述的方便, 通篇引入下列记号:

$$g(x) = \nabla f(x), \quad g_k = \nabla f(x_k), \quad G(x) = \nabla^2 f(x), \quad G_k = \nabla^2 f(x_k).$$

定理 6 (**一阶必要条件**) 设 $f(x)$ 在开集 D 上一阶连续可微. 若 $x^* \in D$ 是 (1.17) 的一个局部极小点, 则必有 $g(x^*) = 0$.

证 取 $x = x^* - \alpha g(x^*) \in D$, 其中 $\alpha > 0$ 为某个常数. 则有

$$\begin{aligned} f(x) &= f(x^*) + g(x^*)^T(x - x^*) + o(\|x - x^*\|) \\ &= f(x^*) - \alpha g(x^*)^T g(x^*) + o(\alpha) \\ &= f(x^*) - \alpha\|g(x^*)\|^2 + o(\alpha). \end{aligned}$$

注意到 $f(x) \geq f(x^*)$ 及 $\alpha > 0$, 我们有

$$0 \leq \|g(x^*)\|^2 \leq \frac{o(\alpha)}{\alpha}.$$

上式两边令 $\alpha \to 0$ 得 $\|g(x^*)\| = 0$, 即 $g(x^*) = 0$. □

定理 7 (**二阶必要条件**) 设 $f(x)$ 在开集 D 上二阶连续可微. 若 $x^* \in D$ 是 (1.17) 的一个局部极小点, 则必有 $g(x^*) = 0$ 且 $G(x^*)$ 是半正定矩阵.

证 设 x^* 是一局部极小点, 那么由定理 6 可知 $g(x^*) = 0$. 下面只需证明 $G(x^*)$ 的半正定性. 任取 $x = x^* + \alpha d \in D$, 其中 $\alpha > 0$ 且 $d \in \mathbb{R}^n$. 由泰勒展开式得

$$0 \leq f(x) - f(x^*) = \frac{1}{2}\alpha^2 d^T G(x^*) d + o(\alpha^2),$$

即

$$d^T G(x^*) d + \frac{o(2\alpha^2)}{\alpha^2} \geq 0.$$

对上式令 $\alpha \to 0$, 即得 $d^T G(x^*) d \geq 0$, 从而定理成立. □

定理 8 (**二阶充分条件**) 设 $f(x)$ 在开集 D 上二阶连续可微. 若 $x^* \in D$ 满足条件 $g(x^*) = 0$ 及 $G(x^*)$ 是正定矩阵, 则 x^* 是 (1.17) 的一个局部极小点.

证 任取 $x = x^* + \alpha d \in D$, 其中 $\alpha > 0$ 且 $d \in \mathbb{R}^n$. 由泰勒公式得

$$f(x^* + \alpha d) = f(x^*) + g(x^*)^T d + \frac{1}{2}\alpha^2 d^T G(x^* + \theta\alpha d) d,$$

其中 $\theta \in (0, 1)$. 注意到 $g(x^*) = 0$, $G(x^*)$ 正定和 f 二阶连续可微, 故存在 $\delta > 0$, 使得 $G(x^* + \theta\alpha d)$ 在$\|\theta\alpha d\| < \delta$ 范围内正定. 因此由上式即得

$$f(x^* + \alpha d) > f(x^*),$$

从而定理成立. □

一般来说, 目标函数的稳定点不一定是极小点. 但对于目标函数是凸函数的无约束优化问题, 其稳定点、局部极小点和全局极小点三者是等价的.

定理 9 设 $f(x)$ 在 $\mathbb{R}^n$ 上是凸函数并且是一阶连续可微的. 则 $x^* \in \mathbb{R}^n$ 是 (1.17) 的全局极小点的充要条件是 $g(x^*) = 0$.

证 只需证明充分性, 必要性是显然的. 设 $g(x^*) = 0$. 由凸函数的判别定理 4(1), 可得

$$f(x) \geq f(x^*) + g(x^*)^T(x - x^*) = f(x^*), \quad \forall\, x \in \mathbb{R}^n,$$

这表明 x^* 是全局极小点. □

1.6 无约束优化问题的算法框架

在数值优化中, 一般采用迭代法求解无约束优化问题

$$\min f(x) \tag{1.18}$$

的极小点. 迭代法的基本思想是: 给定一个初始点 x_0, 按照某一迭代规则产生一个迭代序列 $\{x_k\}$. 使得若该序列是有限的, 则最后一个点就是问题 (1.18) 的极小点; 否则, 若序列 $\{x_k\}$ 是无穷点列时, 它有极限点且这个极限点即为问题 (1.18) 的极小点.

设 x_k 为第 k 次迭代点, d_k 为第 k 次搜索方向, α_k 为第 k 次步长因子, 则第 k 次迭代完成后可得到新一轮 (第 $k+1$ 次) 的迭代点

$$x_{k+1} = x_k + \alpha_k d_k. \tag{1.19}$$

因此, 我们可以写出求解无约束优化问题 (1.18) 的一般算法框架如下.

算法 1 (**无约束问题的一般算法框架**)

步 0 给定初始化参数及初始迭代点 x_0. 置 $k := 0$.

步 1 若 x_k 满足某种终止准则, 停止迭代, 以 x_k 作为近似极小点.

步 2 通过求解 x_k 处的某个子问题确定下降方向 x_k.

步 3 通过某种搜索方式确定步长因子 α_k, 使得 $f(x_k + \alpha_k d_k) < f(x_k)$.

步 4 令 $x_{k+1} := x_k + \alpha_k d_k$, $k := k + 1$, 转步 1.

为了方便, 通常称上述算法中的 $s_k=\alpha_k d_k$ 为第 k 次迭代的位移. 从算法 1 可以看出, 不同的位移 (不同的搜索方向及步长因子) 即产生了不同的迭代算法. 为了保证算法的收敛性, 一般要求搜索方向为所谓的下降方向:

定义 10 若存在 $\bar{\alpha}>0$, 使得对任意的 $\alpha\in(0,\bar{\alpha}]$ 和 $d_k\neq 0$, 有

$$f(x_k+\alpha d_k)<f(x_k).$$

则称 d_k 为 $f(x)$ 在 x_k 处的一个下降方向.

若目标函数 f 是一阶连续可微的, 则判别 d_k 是否为下降方向将有更为方便的判别条件.

引理 2 设函数 $f:D\subset\mathbb{R}^n\to\mathbb{R}$ 在开集 D 上一阶连续可微, 则 d_k 为 $f(x)$ 在 x_k 处一个下降方向的充要条件是

$$\nabla f(x_k)^T d_k<0. \tag{1.20}$$

证 由泰勒展开式得

$$f(x_k+\alpha d_k)=f(x_k)+\alpha\nabla f(x_k)^T d_k+o(\alpha).$$

若 (1.20) 成立, 则对于充分小的 $\alpha>0$, 显然有 $f(x_k+\alpha d_k)<f(x_k$, 即 d_k 为 $f(x)$ 在 x_k 处的一个下降方向. 反之, 则有

$$\alpha\nabla f(x_k)^T d_k+o(\alpha)<0,$$

即

$$\nabla f(x_k)^T d_k+\frac{o(\alpha)}{\alpha}<0.$$

由于 $\lim\limits_{\alpha\to 0}\dfrac{o(\alpha)}{\alpha}=0$, 故可推得 (1.20) 成立. □

算法的收敛性是设计一个迭代算法的最起码要求. 关于收敛性, 有所谓的 “局部收敛性” 和 “全局收敛性” 概念:

定义 11 若某算法只有当初始点 x_0 充分接近极小点 x^* 时, 由算法产生的点列 $\{x_k\}$ 才收敛于 x^*, 则称该算法具有局部收敛性. 若对于任意的初始点 x_0, 由算法产生的点列 $\{x_k\}$ 都收敛于 x^*, 则称该算法具有全局收敛性.

算法的局部收敛速度是衡量一个算法好坏的重要指标, 我们给出有关收敛速度的概念如下.

定义 12 设算法产生的点列 $\{x_k\}$ 收敛于极小点 x^*, 且

$$\lim_{k\to\infty}\frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|^p}=\theta.$$

(1) 若 $p=1$ 且 $0<\theta<1$, 则称该算法具有线性收敛速度 (或线性收敛的).

(2) 若 $p=1$ 且 $\theta=0$, 则称该算法具有超线性收敛速度 (或超线性收敛的).

(3) 若 $p=2$ 且 $0<\theta<\infty$, 则称该算法具有平方收敛速度 (或平方收敛的).

(4) 一般地, 若 $p>2$ 且 $0<\theta<\infty$, 则称该算法具有 p 阶收敛速度 (或 p 阶收敛的).

在计算机上实现一个迭代法, 通常需要一个迭代终止条件. 常用的基本终止条件有下列三种:

(1) 位移的绝对误差或相对误差充分小, 即

$$\|x_{k+1}-x_k\|<\varepsilon,\quad \text{或}\quad \frac{\|x_{k+1}-x_k\|}{\|x_k\|}<\varepsilon,$$

其中 ε 是充分小的正数.

(2) 目标函数的绝对误差或相对误差充分小, 即

$$|f(x_{k+1})-f(x_k)|<\varepsilon,\quad \text{或}\quad \frac{|f(x_{k+1})-f(x_k)|}{|f(x_k)|}<\varepsilon,$$

其中 ε 是充分小的正数.

(3) 目标函数的梯度的范数充分小, 即

$$\|\nabla f(x_k)\|<\varepsilon,$$

其中 ε 是充分小的正数.

习 题 1

1. 验证下列各集合是凸集:

(1) $S=\{(x_1,x_2)|2x_1+x_2\geq 1, x_1-2x_2\geq 1\}$;

(2) $S=\{(x_1,x_2)|x_1^2+y_2^2\leq 1\}$;

(3) $S=\{(x_1,x_2)||x_2|<x_1\}$.

2. 判断下列函数为凸 (凹) 函数或严格凸 (凹) 函数:

(1) $f(x)=x_1^2+2x_2^2$;

(2) $f(x)=3x_1^2-6x_1x_2+x_2^2$;

(3) $f(x)=x_1^2-2x_1x_2+x_2^2+2x_1+3x_2$;

(4) $f(x)=2x_1^2+x_2^2+2x_3^2+x_1x_2-3x_1x_3+x_1-x_3$.

3. 证明: $f(x)=\dfrac{1}{2}x^TGx+b^Tx$ 为严格凸函数当且仅当 Hessian 矩阵 G 正定.

4. 若对任意 $x\in\mathbb{R}^n$ 及实数 $\theta>0$ 都有 $f(\theta x)=\theta(x)$, 证明: $f(x)$ 在 $\mathbb{R}^n$ 上为凸函数的充要条件是 $\forall x,y\in\mathbb{R}^n$, $f(x+y)\le f(x)+f(y)$.

5. 设 $f:\mathbb{R}^3\to\mathbb{R}$ 定义为: $f(x)=(x_1+3x_2)^2+2(x_1-x_3)^4+(x_2-2x_3)^4$. 证明: $x^*=(0,0,0)^T$ 是 $f(x)$ 的稳定点, 且 x^* 是 $f(x)$ 在 $\mathbb{R}^3$ 上的严格全局极小点.

6. 设 $f:\mathbb{R}^n\to\mathbb{R}$ 是凸函数, $x^{(1)},\cdots,x^{(m)}\in\mathbb{R}^n$, $\lambda_1,\cdots,\lambda_m$ 是非负实数且 $\lambda_1+\cdots+\lambda_m=1$. 证明: $f(\lambda_1x^{(1)})+\cdots+\lambda_mx^{(m)}\le\lambda_1f(x^{(1)})+\cdots+\lambda_mf(x^{(m)})$.

7. 设 f, g 都是 $\mathbb{R}^n$ 上的凸函数, 证明 $f+g$ 及 $\max\{|f|,|g|\}$ 也是凸函数.

8. 设 $f(x)=x_1^2+x_2^3-\dfrac{1}{3}x_1^3$, 证明 f 有严格局部极小点 $(0,0)^T$ 和鞍点 $(1,0)$.

9. 设三个序列 $x_k=\left(\dfrac{1}{2}\right)^{2^k}$, $y_k=\mathrm{e}^{-k\ln k}$, $z_k=\dfrac{1}{k^2}$, 当 $k\to\infty$ 时, 它们均收敛于 0. 试证 $\{x_k\}$ 二阶收敛, $\{y_k\}$超线性收敛, $\{z_k\}$ 线性收敛.

10. 设 $f(x_1,x_2)=10-2(x_1^2-x_2)^2$.

$$S=\{(x_1,x_2)|-11\le x_1\le 1,-1\le x_2\le 1\}.$$

$f(x_1,x_2)$ 是否为 S 上的凸函数?

第二章　线搜索技术

从本章开始, 介绍无约束优化问题的一些常用数值方法. 我们考虑下面的无约束优化模型

$$\min_{x\in\mathbb{R}^n} f(x).$$

众所周知, 研究上述无约束优化问题的数值方法, 不仅是出于实际问题的需要, 同时也是研究约束优化问题数值方法的基础. 本章主要讨论一维线搜索算法及其收敛性分析.

我们在第 1 章论及无约束优化问题迭代算法的一般框架时, 其中有下面的一个迭代步:

通过某种搜索方式确定步长因子 α_k, 使得

$$f(x_k+\alpha_k d_k) < f(x_k). \tag{2.1}$$

这实际上是 (n 个变量的) 目标函数 $f(x)$ 在一个规定的方向上移动所形成的单变量优化问题, 也就是所谓的“线搜索”或“一维搜索”技术. 令

$$\phi(\alpha) = f(x_k+\alpha d_k), \tag{2.2}$$

这样, 搜索式 (2.1) 等价于求步长 α_k 使得

$$\phi(\alpha_k) < \phi(0).$$

线搜索有精确线搜索和非精确线搜索之分. 所谓精确线搜索, 是指求 α_k 使目标函数 f 沿方向 d_k 达到极小, 即

$$f(x_k+\alpha_k d_k) = \min_{\alpha>0} f(x_k+\alpha d_k),$$

或

$$\phi(\alpha_k) = \min_{\alpha>0} \phi(\alpha).$$

若 $f(x)$ 是连续可微的, 那么由精确线搜索得到的步长因子 α_k 具有如下性质:

$$\nabla f(x_k+\alpha_k d_k)^T d_k = 0 \text{（亦即 } g_{k+1}^T d_k = 0\text{）}. \tag{2.3}$$

上述性质在后面的算法收敛性分析中将起着重要的作用.

所谓非精确线搜索, 是指选取 α_k 使目标函数 f 得到可接受的下降量, 即 $\Delta f_k = f(x_k) - f(x_k+\alpha_k d_k) > 0$ 是可接受的.

精确线搜索的基本思想是: 首先确定包含问题最优解的搜索区间, 然后采用某种插值或分割技术缩小这个区间, 进行搜索求解. 下面给出搜索区间的定义.

定义 13 设 ϕ 是定义在实数集上一元实函数, $\alpha^* \in [0,+\infty)$, 并且

$$\phi(\alpha^*) = \min_{\alpha\geq 0} \phi(\alpha). \tag{2.4}$$

若存在区间 $[a,b] \subset [0,+\infty)$, 使 $\alpha^* \in (a,b)$, 则称 $[a,b]$ 是极小化问题 (2.4) 的搜索区间. 进一步, 若 α^* 使得 $\phi(\alpha)$ 在 $[a,\alpha^*]$ 上严格递减, 在 $[\alpha^*,b]$ 上严格递增, 则称 $[a,b]$ 是 $\phi(\alpha)$ 的单峰区间, $\phi(\alpha)$ 是 $[a,b]$ 上的单峰函数.

下面介绍一种确定搜索区间并保证具有近似单峰性质的数值算法—进退法, 其基本思想是从一点出发, 按一定步长, 试图确定函数值呈现“高-低-高”的三点, 从而得到一个近似的单峰区间.

算法 2 (进退法)

步 1　选取 $\alpha_0 \geq 0$, $h_0 > 0$. 计算 $\phi_0 := \phi(\alpha_0)$. 置 $k := 0$.

步 2　令 $\alpha_{k+1} = \alpha_k + h_k$, 计算 $\phi_{k+1} := \phi(\alpha_{k+1})$. 若 $\phi_{k+1} < \phi_k$, 转步 3, 否则转步 4.

步 3　加大步长. 令 $h_{k+1} := 2h_k$, $\alpha := \alpha_k$, $\alpha_k := \alpha_{k+1}$, $\phi_k := \phi_{k+1}$, $k := k+1$, 转步 2.

步 4　反向搜索或输出. 若 $k = 0$, 令 $h_1 := h_0$, $\alpha := \alpha_1$, $\alpha_1 := \alpha_0$, $\phi_1 := \phi_0$, $k := 1$, 转步 2; 否则停止迭代, 令

$$a = \min\{\alpha, \alpha_{k+1}\}, \quad a = \max\{\alpha, \alpha_{k+1}\}.$$

输出 $[a,b]$.

2.1　精确线搜索及其 Matlab 实现

精确线搜索分为两类. 一类是使用导数的搜索, 如插值法, 牛顿法及抛物线法等; 另一类是不用导数的搜索, 如 0.618 法, 分数法及成功-失败法等. 本书仅介绍 0.618 法和二次插值逼近法.

1. 黄金分割法

黄金分割法也称为 0.618 法, 其基本思想是通过试探点函数值得比较, 是包含极小点的搜索区间不断缩小. 该方法仅需要计算函数值, 适用范围广, 使用方便. 下面我们来推导 0.618 法的计算公式.

设

$$\phi(s) = f(x_k + sd_k),$$

其中 $\phi(s)$ 是搜索区间 $[a_0, b_0]$ 上的单峰函数. 在第 i 次迭代时搜索区间为 $[a_i, b_i]$. 取两个试探点为 $p_i, q_i \in [a_i, b_i]$ 且 $p_i < q_i$. 计算 $\phi(p_i$ 和 $\phi(q_i)$. 根据单峰函数的性质, 可能会出现如下两种情形之一:

(1) 若 $\phi(p_i) \le \phi(q_i)$, 则令 $a_{i+1} := a_i$, $b_{i+1} := q_i$;

(2) 若 $\phi(p_i) > \phi(q_i)$, 则令 $a_{i+1} := p_i$, $b_{i+1} := b_i$.

我们要求两个试探点 p_i 和 q_i 满足下述两个条件:

(a) $[a_i, q_i]$ 与 $[p_i, b_i]$ 的长度相同, 即 $b_i - p_i = q_i - a_i$;

(b) 区间长度的缩短率相同, 即 $b_{i+1} - a_{i+1} = t(b_i - a_i)$.

从而可得

$$p_i = a_i + (1-t)(b_i - a_i), \quad q_i = a_i + t(b_i - a_i). \tag{2.5}$$

现在考虑情形 (1). 此时, 新的搜索区间为

$$[a_{i+i}, b_{i+1}] = [a_i, q_i].$$

为了进一步缩短搜索区间, 需取新的试探点 p_{i+1}, q_{i+1}. 由 (2.5) 得

$$\begin{aligned} q_{i+1} &= a_{i+1} + t(b_{i+1} - a_{i+1}) \\ &= a_i + t(q_i - a_i) \\ &= a_i + t^2(b_i - a_i). \end{aligned}$$

若令

$$t^2 = 1 - t, \quad t > 0, \tag{2.6}$$

则

$$q_{i+1} = a_i + (1-t)(b_i - a_i) = p_i.$$

这样, 新的试探点 q_{i+1} 就不需要重新计算. 类似地, 对于情形 (2), 也有相同的结论.

解方程 (2.6) 得区间长度缩短率为

$$t = \frac{\sqrt{5}-1}{2} \approx 0.618.$$

因此, 我们可以写出 0.618 法的计算步骤如下.

算法 3 (0.618 法)

步 0 确定初始搜索区间 $[a_0, b_0]$ 和容许误差 $\varepsilon > 0$. 计算初始试探点

$$p_0 = a_0 + 0.382(b_0 - a_0), \quad q_0 = a_0 + 0.618(b_0 - a_0)$$

及相应的函数值 $\phi(p_0)$, $\phi(q_0)$. 置 $i := 0$.

步 1 若 $\phi(p_i) \le \phi(q_i)$, 转步 2; 否则, 转步 3.

步 2 计算左试探点. 若 $|q_i - a_i| \le \varepsilon$, 停算, 输出 p_i. 否则, 令

$$\begin{aligned} &a_{i+1} := a_i, \quad b_{i+1} := q_i, \quad \phi(q_{i+1}) := \phi(p_i), \\ &q_{i+1} := p_i, \quad p_{i+1} := a_{i+1} + 0.382(b_{i+1} - a_{i+}). \end{aligned}$$

计算 $\phi(p_{i+1})$, $i := i+1$, 转步 1.

步 3 计算右试探点. 若 $|b_i - p_i| \le \varepsilon$, 停算, 输出 q_i. 否则, 令

$$\begin{aligned} &a_{i+1} := p_i, \quad b_{i+1} := b_i, \quad \phi(p_{i+1}) := \phi(q_i), \\ &p_{i+1} := q_i, \quad q_{i+1} := a_{i+1} + 0.618(b_{i+1} - a_{i+}). \end{aligned}$$

计算 $\phi(q_{i+1})$, $i := i+1$, 转步 1.

值得说明的是, 由于每次迭代搜索区间的收缩率是 $t = 0.618$, 故 0.618 法只是线性收敛的, 即这一方法的计算效率并不高. 但该方法每次迭代只需计算一次函数值的优点足以弥补这一缺憾.

下面给出用 0.618 法求单变量函数 ϕ 在单峰区间上近似极小点的 Matlab 程序.

程序 1 (0.618 法程序) 用 0.618 法求单变量函数 ϕ 在单峰区间 $[a,b]$ 上的近似极小点.

```
function [s,phis,k,G,E]=golds(phi,a,b,delta,epsilon)
%输入: phi是目标函数, a, b 是搜索区间的两个端点
%      delta, epsilon分别是自变量和函数值的容许误差
%输出: s, phis分别是近似极小点和极小值, G是nx4矩阵,
%      其第k行分别是a,p,q,b的第k次迭代值[ak,pk,qk,bk],
%      E=[ds,dphi], 分别是s和phis的误差限.
t=(sqrt(5)-1)/2;  h=b-a;
phia=feval(phi,a); phib=feval(phi,b);
p=a+(1-t)*h;  q=a+t*h;
phip=feval(phi,p); phiq=feval(phi,q);
k=1;  G(k,:)=[a, p, q, b];
while(abs(phib-phia)>epsilon)|(h>delta)
    if(phip<phiq)
        b=q;  phib=phiq; q=p; phiq=phip;
        h=b-a; p=a+(1-t)*h; phip=feval(phi,p);
    else
        a=p; phia=phip; p=q; phip=phiq;
        h=b-a;  q=a+t*h;  phiq=feval(phi,q);
    end
    k=k+1;  G(k,:)=[a, p, q, b];
end
ds=abs(b-a); dphi=abs(phib-phia);
if(phip<=phiq)
    s=p;  phis=phip;
else
    s=q;  phis=phiq;
end
E=[ds,dphi];
```

例 8 用 0.618 法程序 1 求函数 $\phi(x)=x^2-\sin(x)$ 在 $[0,1]$ 上的极小点. 取容许误差 $\delta=10^{-4}$, $\varepsilon=10^{-5}$.

解 在命令窗口输入如下命令

```
[s,phis,k,G,E]=golds(inline('s^2-sin(s)'),0,1,1e-4,1e-5)
```

回车后即得如下数值结果:

表 3.1 用 0.618 法求单变量函数极小点的数值结果.

迭代次数	近似极小点 ($\bar{s}$)	$\lvert b_k - a_k \rvert$ 的值	$\lvert \phi(b_k) - \phi(a_k) \rvert$ 的值
21	0.450183	6.6107×10^{-5}	1.1075×10^{-9}

2. 抛物线法

抛物线法也叫二次插值法, 其基本思想是: 在搜索区间中不断地使用二次多项式去近似目标函数, 并逐步用插值多项式的极小点去逼近先搜索问题

$$\min_{s>0} \phi(s) = f(x_k + sd_k)$$

的极小点. 下面我们详细介绍这一方法.

设已知三点

$$s_0, \quad s_1 = s_0 + h, \quad s_2 = s_0 + 2h, \quad (h > 0)$$

处的函数值 ϕ_0, ϕ_1, ϕ_2 且满足

$$\phi_1 < \phi_0, \quad \phi_1 < \phi_2.$$

上述条件保证了函数 ϕ 在区间 $[s_0, s_2]$ 上是单峰函数. 则满足上述条件的二次 Lagrabge 插值多项式为

$$q(s) = \frac{(s-s_1)(s-s_2)}{2h^2}\phi_0 - \frac{(s-s_0)(s-s_2)}{h^2}\phi_1 + \frac{(s-s_0)(s-s_1)}{2h^2}\phi_2.$$

$q(s)$ 的一阶导数为

$$q'(s) = \frac{2s-s_1-s_2}{2h^2}\phi_0 - \frac{2s-s_0-s_2}{h^2}\phi_1 + \frac{2s-s_0-s_1}{2h^2}\phi_2. \tag{2.7}$$

令 $q'(s) = 0$ 解得

$$\begin{aligned} \bar{s} &= \frac{(s_1+s_2)\phi_0 - 2(s_0+s_2)\phi_1 + (s_0+s_1)\phi_2}{2(\phi_0 - 2\phi_1 + \phi_2)} \\ &= \frac{(2s_0+3h)\phi_0 - 2(2s_0+2h)\phi_1 + (2s_0+h)\phi_2}{2(\phi_0 - 2\phi_1 + \phi_2)} \\ &= s_0 + \frac{(3\phi_0 - 4\phi_1 + \phi_2)h}{2(\phi_0 - 2\phi_1 + \phi_2)} := s_0 + \bar{h}, \end{aligned} \tag{2.8}$$

这里

$$\bar{h} = \frac{(4\phi_1 - 3\phi_0 - \phi_2)h}{2(2\phi_1 - \phi_0 - \phi_2)}. \tag{2.9}$$

又 $q(s)$ 的二阶导数分别为

$$q''(s) = \frac{\phi_0}{h^2} - \frac{2\phi_1}{h^2} + \frac{\phi_2}{h^2} = \frac{\phi_0 - 2\phi_1 + \phi_2}{h^2} > 0.$$

故 $q(s)$ 为凸二次函数, 从而 $s_{\min}$ 是 $q(s)$ 的全局极小点.

注意到, $\bar{s} = s_0 + \bar{h}$ 比 s_0 更好地逼近 s^*. 故可用 $\bar{s}$, $\bar{h}$ 分别替换 s_0 和 h 并重复上述计算过程, 求出新的 $\bar{s}$ 和新的 $\bar{h}$. 重复这一迭代过程, 直到得到所需的精度为止. 值得说明的是, 这一算法中目标函数的导数在 (2.7) 中隐式地用来确定用来确定二次插值多项式的极小点, 而算法的程序实现中并不需要使用导数值.

算法 4 (抛物线法)

步 0 由算法 2 确定三点 $s_0 < s_1 < s_2$, 对应的函数值 ϕ_0, ϕ_1, ϕ_2 满足

$$\phi_1 < \phi_0, \quad \phi_1 < \phi_2.$$

设定容许误差 $\varepsilon > 0$.

步 1 若 $|s_2 - s_0| < \varepsilon$, 停算, 输出 $s^* \approx s_1$.

步 2 计算插值点. 根据 (2.8) 计算 $\bar{s}$ 和 $\bar{\phi} := \phi(\bar{s})$. 若 $\phi_1 \le \bar{\phi}$, 转步 4, 否则, 转步 3.

步 3 若 $s_1 > \bar{s}$, 则 $s_2 := s_1$, $s_1 := \bar{s}$, $\phi_2 := \phi_1$, $\phi_1 := \bar{\phi}$, 转步 1. 否则, $s_0 := s_1$, $s_1 := \bar{s}$, $\phi_0 := \phi_1$, $\phi_1 := \bar{\phi}$, 转步 1.

步 4 若 $s_1 < \bar{s}$, 则 $s_2 := \bar{s}$, $\phi_2 := \bar{\phi}$, 转步 1. 否则, $s_0 := \bar{s}$, $\phi_0 := \bar{\phi}$, 转步 1.

下面给出用抛物法求单变量函数 ϕ 在单峰区间上近似极小点的 Matlab 程序.

程序 2 (抛物线法程序) 求函数 $\phi(s)$ 在区间 $[a, b]$ 上的局部极小值, 从初始点 s_0 开始, 然后在区间 $[a, s_0]$ 和 $[s_0, b]$ 上进行搜索.

```
function [s,phis,ds,dphi,S]=qmin(phi,a,b,delta,epsilon)
%输入: phi 是目标函数, a和b是搜索区间的端点
%       delta,epsilon是容许误差
%输出: s是近似极小点, phis是对应的近似极小值; k是迭代次数
%       ds是迭代终止时的步长, dphi是|phi(s1)-phi(s)|; S是迭代向量
```

```
s0=a; maxj=20; maxk=30; big=1e6; err=1; k=1;
S(k)=s0; cond=0;  h=1; ds=0.00001;
if (abs(s0)>1e4), h=abs(s0)*(1e-4); end
while (k<maxk & err>epsilon &cond~=5)
    f1=(feval(phi,s0+ds)-feval(phi,s0-ds))/(2*ds);
    if(f1>0), h=-abs(h); end
    s1=s0+h;     s2=s0+2*h;      bars=s0;
    phi0=feval(phi,s0);  phi1=feval(phi,s1);
    phi2=feval(phi,s2);  barphi=phi0; cond=0;
    j=0;  %确定h使得phi1<phi0且phi1<phi2
    while(j<maxj&abs(h)>delta&cond==0)
        if (phi0<=phi1),
            s2=s1; phi2=phi1; h=0.5*h;
            s1=s0+h; phi1=feval(phi,s1);
        else if (phi2<phi1),
                s1=s2; phi1=phi2; h=2*h;
                s2=s0+2*h; phi2=feval(phi,s2);
            else
                cond=-1;
            end
        end
        j=j+1;
        if(abs(h)>big|abs(s0)>big), cond=5; end
    end
    if(cond==5)
        bars=s1; barphi=feval(phi,s1);
    else
        %二次插值求phis
        d=2*(2*phi1-phi0-phi2);
        if(d<0),
            barh=h*(4*phi1-3*phi0-phi2)/d;
        else
            barh=h/3; cond=4;
        end
        bars=s0+barh;   barphi=feval(phi,bars);
        h=abs(h);  h0=abs(barh);
        h1=abs(barh-h);  h2=abs(barh-2*h);
```

```
        %确定下一次迭代的h值
        if(h0<h), h=h0; end
        if(h1<h), h=h1; end
        if(h2<h), h=h2; end
        if(h==0), h=barh; end
        if(h<delta), cond=1; end
        if(abs(h)>big|abs(bars)>big), cond=5; end
        err=abs(phi1-barphi);
        s0=bars;  k=k+1; S(k)=s0;
    end
    if(cond==2&h<delta), cond=3; end
end
s=s0;  phis=feval(phi,s);
ds=h;  dphi=err;
```

例 9 用二次插值法程序 2 求函数 $\phi(x)=x^2-\sin(x)$ 在 $[0,1]$ 上的极小点. 取容许误差 $\delta=10^{-4}$, $\varepsilon=10^{-6}$.

解 在命令窗口输入如下命令

`[s,phis,k,ds,dphi,S]=qmin(inline('s^2-sin(s)'),0,1,1e-4,1e-6)`

回车后即得如下数值结果:

表 3.1 用二次插值法求单变量函数极小点的数值结果.

迭代次数	近似极小点 ($\bar{s}$)	$\lvert\bar{s}-s_1\rvert$ 的值	$\lvert\phi(\bar{s})-\phi(s_1)\rvert$ 的值
4	0.450184	1.5340×10^{-5}	1.6984×10^{-9}

2.2　非精确线搜索及其 Matlab 实现

线搜索技术是求解许多优化问题下降算法的基本组成部分, 但精确线搜索往往需要计算很多的函数值和梯度值, 从而耗费较多的计算资源. 特别是当迭代点远离最优点时, 精确线搜索通常不是十分有效和合理的. 对于许多优化算法, 其收敛速度并不依赖于精确搜索过程. 因此, 既能保证目标函数具有可接受的下降量又能使最终形成的迭代序列收敛的非精确线搜索变得越来越流行. 本书着重介绍非精确线搜索中的 Wolfe 准则和 Armijo 准则.

1. Wolfe 准则

Wolfe 准则是指: 给定 $\rho \in (0, 0.5)$, $\sigma \in (\rho, 0.5)$, 求 α_k 使得下面两个不等式同时成立:

$$f(x_k + \alpha_k d_k) \le f(x_k) + \rho \alpha_k g_k^T d_k, \tag{2.10}$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \ge \sigma g_k^T d_k, \tag{2.11}$$

其中 $g_k = g(x_k) = \nabla f(x_k)$. 条件 (2.11) 有时也用另一个更强的条件

$$|\nabla f(x_k + \alpha_k d_k)^T d_k| \le -\sigma g_k^T d_k \tag{2.12}$$

来代替. 这样, 当 $\sigma > 0$ 充分小时, 可保证 (2.12) 变成近似精确线搜索. (2.10) 和 (2.12) 也称为强 Wolfe 准则.

强 Wolfe 准则表明, 由该准则得到的新的迭代点 $x_{k+1} = x_k + \alpha_k d_k$ 在 x_k 的某一邻域内且使目标函数值有一定的下降量.

由于 $g_k^T d_k < 0$, 可以证明 Wolfe 准则的有限终止性, 即步长 α_k 的存在性. 我们有下面的定理.

定理 10 设 $f(x)$ 有下界且 $g_k^T d_k < 0$, 令 $\rho \in (0, 0.5)$, $\sigma \in (\rho, 1)$. 则存在一个区间 $[a, b]$, $0 < a < b$, 使每个 $\alpha \in [a, b]$ 均满足 (2.10) 和 (2.12).

2. Armijo 准则

Armijo 准则是指: 给定 $\beta \in (0, 1)$, $\sigma \in (0, 0.5)$. 令步长因子 $\alpha_k = \beta^{m_k}$, 其中 m_k 是满足下列不等式的最小非负整数:

$$f(x_k + \beta^m d_k) \le f(x_k) + \sigma \beta^m g_k^T d_k. \tag{2.13}$$

可以证明, 若 $f(x)$ 是连续可微的且满足 $g_k^T d_k < 0$, 则 Armijo 准则是有限终止的, 即存在正数 σ, 使得对于充分大的正整数 m, (2.13) 式成立.

为了程序实现的方便, 我们将 Armijo 准则写成下列详细的算法步骤.

算法 5 (**Armijo 准则**)

步 0 给定 $\beta \in (0, 1)$, $\sigma \in (0, 0.5)$. 令 $m := 0$.

步 1 若不等式

$$f(x_k + \beta^m d_k) \le f(x_k) + \sigma \beta^m g_k^T d_k$$

成立, 置 $m_k := m$, $x_{k+1} := x_k + \beta^{m_k} d_k$, 停算. 否则, 转步 2.

步 2 令 $m := m + 1$, 转步 1.

下面给出 Armijo 准则的 Matlab 程序.

程序 3 (**Armijo 准则程序**) Armijo 搜索规则是许多非线性优化算法都必须执行的步骤, 把它编制成可重复利用的程序模块是很有意义的.

```
function mk=armijo(xk,dk )
beta=0.5;  sigma=0.2;
m=0; mmax=20;
while (m<=mmax)
    if(fun(xk+beta^m*dk)<=fun(xk)+sigma*beta^m*gfun(xk)'*dk)
        mk=m;  break;
    end
    m=m+1;
end
alpha=beta^mk
newxk=xk+alpha*dk
fk=fun(xk)
newfk=fun(newxk)
```

说明 程序 3 中 fun 和 gfun 分别是指目标函数和它的梯度函数的子程序. 执行上述过程时这两个子程序必须事先准备好.

我们用上述程序来求解下面的问题.

例 10 考虑无约束优化问题

$$\min_{x\in\mathbb{R}^2} f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2,$$

设当前迭代点 $x_k = (-1, 1)^T$, 下降方向 $d_k = (1, -2)^T$. 利用程序 3 求步长因子 α_k.

解 首先编写好目标函数及其梯度两个 m 文件 fun.m 和 gfun.m:

```
% 目标函数
function f=fun(x)
f=100*(x(1)^2-x(2))^2+(x(1)-1)^2;
% 梯度
function gf=gfun(x)
gf=[400*x(1)*(x(1)^2-x(2))+2*(x(1)-1), -200*(x(1)^2-x(2))]';
```

然后在 Matlab 命令窗口输入下列代码

```
xk=[-1,1]'; dk=[1,-2]'; mk=armijo(xk,dk)
```

可得

$$m_k = 2;\ \ \alpha_k = 0.25;\ \ x_{k+1} = (-0.75, 0.5)^T;\ \ f(x_k) = 4;\ \ f(x_{k+1}) = 3.4531.$$

2.3　线搜索法的收敛性

下面我们给出线搜索法的收敛性结果. 所谓“线搜索法”是指用线搜索技术求步长因子的无约束优化问题下降类算法的简称. 其一般的算法框架是:

算法 6 (线搜索法算法框架)

步 0 初始化. 选取有关参数及初始迭代点 $x_0 \in \mathbb{R}^n$. 设定容许误差 $\varepsilon \ll 1$. 令 $k := 0$.

步 1 检验终止判别准则. 计算 $g_k = \nabla f(x_k)$. 若 $\|g_k\| \le \varepsilon$, 输出 $x^* \approx x_k$, 停算.

步 2 确定下降方向 d_k, 使满足 $g_k^T d_k < 0$.

步 3 确定步长因子 α_k. 可在下列“精确”与“非精确”两种线搜索技术中选用其一:

(1) 用前面介绍的 0.618 法或二次插值法等精确线搜索技术求

$$\alpha_k = \arg\min_{\alpha>0} f(x_k + \alpha d_k). \tag{2.14}$$

(2) 用前面介绍的 Wolfe 准则或 Armijo 准则等非精确线搜索技术求 α_k.

步 4 更新迭代点. 令 $x_{k+1} := x_k + \alpha_k d_k$, $k := k+1$, 转步 1.

值得说明的是, 为了保证这类算法的收敛性, 除了在步 3 要选用适当的线搜索技术外, 搜索方向 d_k 也需满足一定的条件, 即对于所有的 k, d_k 与 $-g_k$ 的夹角 θ_k 满足

$$0 \le \theta_k \le \frac{\pi}{2} - \mu, \quad \mu \in \left(0, \frac{\pi}{2}\right). \tag{2.15}$$

显然, 夹角 θ_k 的余弦为

$$\cos\theta_k = \frac{-g_k^T d_k}{\|g_k\|\|d_k\|}. \tag{2.16}$$

下面的定理描述了基于 Wolfe 准则或 Armijo 准则的非精确线搜索法的收敛性.

定理 11 设 $\{x_k\}$ 是由算法 6 产生的序列, $f(x)$ 有下界且对任意的 $x_0 \in \mathbb{R}^n$, $\nabla f(x)$ 在水平集

$$L(x_0) = \{x \in \mathbb{R}^n |\, f(x) \le f(x_0)\}$$

上存在且一致连续. 若下降方向 d_k 满足条件 (2.15), 则

(1) 采用 Wolfe 准则求搜索步长 α_k 时, 有 $\{\|g_k\|\} \to 0$, $k \to \infty$.

(2) 采用 Armijo 准则求搜索步长 α_k 时, $\{x_k\}$ 的任何聚点 x^* 都满足 $\nabla f(x^*) = 0$.

证 (1) 用反证法. 设存在子列 (仍记指标为 k), 使得 $\|g_k\| \ge \varepsilon > 0$. 注意到 d_k 是下降方向, 由 Wolfe 准则的条件 (2.10) 知, $\{f(x_k)\}$ 是单调下降的. 又 $f(x_k)$ 有下界, 故 $f(x_k)$ 得极限存在, 因此有 $f(x_k) - f(x_{k+1}) \to 0$. 令 $s_k = \alpha_k d_k$, 则由 (2.10) 和 (2.15) 可得 $\cos\theta_k \ge \sin\mu$ 及

$$0 \le -g_k^T(\alpha_k d_k) = -g_k^T s_k \le \frac{1}{\rho}\big(f(x_k) - f(x_{k+1})\big) \to 0. \tag{2.17}$$

故

$$0 \le \|g_k\|\|s_k\| \sin\mu \le \|g_k\|\|s_k\| \cos\theta_k = -g_k^T s_k \to 0.$$

注意到 $\|g_k\| \ge \varepsilon > 0$, 故由上式必有 $\|s_k\| \to 0$. 又由于 $\nabla f(x)$ 在水平集 $L(x_0)$ 上是一致连续的, 我们有

$$\nabla f(x_{k+1})^T s_k = g_k^T s_k + o(\|s_k\|),$$

即

$$\lim_{k\to\infty} \frac{\nabla f(x_{k+1})^T s_k}{g_k^T s_k} = 1,$$

这与 (2.11) 式及 $\sigma < 1$ 矛盾. 因而必有 $\|g_k\| \to 0$.

(2) 用反证法. 假设 x^* 是序列 $\{x_k\}$ 的聚点且 $\nabla f(x^*) \ne 0$. 由定理的条件可得 $f(x_k) \to f(x^*)$ 及 $f(x_k) - f(x_{k+1}) \to 0$. 又由 Armijo 准则的条件 (2.13), 我们有

$$-\sigma g_k^T s_k \to 0, \quad g_k^T s_k \to 0,$$

此处 $s_k = \beta^{m_k} d_k$. 若 $g(x_k) \to 0$ 不成立, 则由上式可得 $\|s_k\| \to 0$. 由于在 Armijo 准则的条件 (2.13) 中, m_k 是使不等式成立的最小非负整数, 因此对于 $\beta^{m_k-1} = \beta^{m_k}/\beta$, 不等式变为

$$f(x_k + \beta^{m_k-1} d_k) - f(x_k) > \sigma\beta^{m_k-1} g_k^T d_k.$$

注意到 $\beta^{m_k-1}d_k = s_k/\beta$, 故上式即

$$f\Big(x_k + \frac{1}{\beta}s_k\Big) - f(x_k) > \sigma g_k^T\Big(\frac{s_k}{\beta}\Big). \tag{2.18}$$

若令 $p_k = \dfrac{s_k}{\|s_k\|}$, 则 $\dfrac{s_k}{\beta} = \dfrac{\|s_k\|}{\beta}p_k$. 由 $\|s_k\| \to 0$ 可知, $\alpha'_k = \dfrac{\|s_k\|}{\beta} \to 0$ 且 (2.18) 式可改写为

$$\frac{f(x_k + \alpha'_k p_k) - f(x_k)}{\alpha'_k} > \sigma g_k^T p_k.$$

因 $\|p_k\| = 1$, 故 $\{\|p_k\|\}$ 有界, 从而存在收敛的子列, 仍记为 $\{\|p_k\|\} \to p^*$, $\|p^*\| = 1$. 对上式两边取极限得

$$\nabla f(x^*)^T p^* \geq \sigma \nabla f(x^*)^T p^*.$$

由此可得

$$\nabla f(x^*)^T p^* \geq 0. \tag{2.19}$$

另一方面, 注意到

$$p_k = \frac{s_k}{\|s_k\|} = \frac{d_k}{\|d_k\|},$$

故有

$$-g_k^T p_k = -g_k^T\Big(\frac{d_k}{\|d_k\|}\Big) = \|g_k\|\cos\theta_k \geq \|g_k\|\sin\mu.$$

对上式取极限得

$$-\nabla f(x^*)^T p^* \geq \|\nabla f(x^*)\|\sin\mu > 0,$$

即

$$\nabla f(x^*)^T p^* < 0,$$

这与 (2.19) 式矛盾. 故必有 $\nabla f(x^*) = 0$. □

下面的定理描述了基于精确线搜索技术的无约束优化问题下降类算法的收敛性.

定理 12　设 $\{x_k\}$ 是由算法 6 产生的序列, $f(x)$ 有下界且对任意的 $x_0 \in \mathbb{R}^n$, $\nabla f(x)$ 在水平集

$$L(x_0) = \{x \in \mathbb{R}^n | \, f(x) \leq f(x_0)\}$$

上存在且一致连续. 若下降方向 d_k 满足条件 (2.15) 且搜索步长 α_k 满足精确线搜索条件 (2.14). 则若非 $g_k = 0$, 必有 $g_k \to 0\,(k \to \infty)$.

证 只需证明 $g_k \neq 0$ 时必有 $g_k \to 0\,(k\to\infty)$ 成立. 用反证法. 若 $g_k \to 0$ 不成立, 则存在常数 $\varepsilon > 0$ 和一个子列 (仍记为该序列本身) 使得 $\|g_k\| \geq \varepsilon$.

首先, 由算法 6 的步 4 知, $\{f(x_k)\}$ 是单调下降的, 再注意到 $f(x_k)$ 有下界, 故序列 $\{f(x_k)\}$ 的极限存在, 从而

$$f(x_{k+1}) - f(x_k) \to 0, \quad k\to\infty. \tag{2.20}$$

另一方面, 我们有

$$\frac{-g_k^T d_k}{\|d_k\|} = \|g_k\|\cos\theta_k \geq \varepsilon\sin\mu \equiv \varepsilon_0. \tag{2.21}$$

由泰勒展开式得

$$\begin{aligned} f(x_k+\alpha d_k) &= f(x_k) + \alpha g(\xi_k)^T d_k \\ &= f(x_k) + \alpha g_k^T d_k + \alpha[g(\xi_k) - g_k]^T d_k \\ &\leq f(x_k) + \alpha\|d_k\|\left[\frac{g_k^T d_k}{\|d_k\|} + \|g(\xi_k) - g_k\|\right], \end{aligned} \tag{2.22}$$

其中 ξ_k 是连接 x_k 与 $x_k+\alpha d_k$ 线段上的某一点. 由于 $g(x) = \nabla f(x)$ 在水平集 $L(x_0)$ 上一致连续, 故存在 $\bar\alpha > 0$, 使当 $0 \leq \alpha\|d_k\| \leq \bar\alpha$ 时,

$$\|g(\xi_k) - g_k\| \leq \frac{1}{2}\varepsilon_0, \quad \forall\, k \geq 0. \tag{2.23}$$

在 (2.22) 中令 $\alpha = \bar\alpha/\|d_k\|$, 并利用 (2.21) 和 (2.23) 得

$$\begin{aligned} f\Big(x_k + \frac{\bar\alpha}{\|d_k\|}d_k\Big) &\leq f(x_k) + \bar\alpha\left[\frac{g_k^T d_k}{\|d_k\|} + \|g(\xi_k) - g_k\|\right] \\ &\leq f(x_k) + \bar\alpha\Big[-\varepsilon_0 + \frac{1}{2}\varepsilon_0\Big] \\ &= f(x_k) - \frac{1}{2}\bar\alpha\varepsilon_0. \end{aligned}$$

由精确搜索条件得

$$f(x_{k+1}) \leq f\Big(x_k + \frac{\bar\alpha}{\|d_k\|}d_k\Big) \leq f(x_k) - \frac{1}{2}\bar\alpha\varepsilon_0.$$

这与 (2.20) 式矛盾, 从而 $g_k \to 0$. □

习 题 2

1. 用 0.618 法求解

$$\min f(x)=x^2-x-1.$$

初始区间为 $[-1,1]$, 区间精度为 $\delta=0.05$.

2. 用 0.618 法求

$$\min f(x)=x^3-2x+1$$

的近似最优解, 初始搜索区间为 $[0,3]$. 区间精度为 $\delta=0.15$.

3. 用抛物线法求 $f(x)=x^2-6x+2$ 的近似极小点, 给定初始点 $x_0=1$, 初始步长 $h_0=0.1$.

4. 用抛物线法法求

$$\min_{x\geq 0} f(x)=x^3-2x+1$$

的近似最优解, 初始搜索区间为 $[0,3]$, 初始插值点 $x_0=1$, 终止条件为 $|x_{k+1}-x_k|<\delta=0.01$.

5. 分别用教材所给的 0.618 法和抛物线法 Matlab 程序计算下列问题的近似最优解:

(1) $\min f(x)=e^{-x}+x^2$;　　(2) $\min f(x)=3x^4-4x^3-12x^2$;

(3) $\min f(x)=x^4+2x+4$;　　(4) $\min f(x)=x^3-3x+1$.

6. 利用教材所给的 Armijo 搜索 Matlab 程序计算下面的问题:

$$\min \varphi(x)=f(x_k+\alpha d_k),$$

其中 $f(x)=100(x_2-x_1^2)^2+(1-x_1)^2$, $x_k=(-1,1)^T$, $d_k=(1,1)^T$.

第三章　最速下降法和牛顿法

本章讨论无约束优化问题

$$\min_{x\in\mathbb{R}^n} f(x) \tag{3.1}$$

的最速下降法和牛顿法及其改进算法. 其中最速下降法是求解无约束优化问题最简单和最古老的方法之一, 虽然时至今日它不再具有实用性, 但它却研究其它无约束优化算法的基础, 许多有效算法都是以它基础通过改进或修正而得到的. 此外, 牛顿法也是一种经典的无约束优化算法, 并且因其收敛速度快以及具有自适应性等优点而至今仍受到科技工作者的青睐.

3.1　最速下降方法及其 Matlab 实现

在第 2 章关于无约束优化问题下降类算法的一般框架时提及, 用不同的方式确定搜索方向或搜索步长, 就会得到不同的算法. 最速下降法是用负梯度方向

$$d_k = -\nabla f(x_k) \tag{3.2}$$

作为搜索方向的 (因此也称为梯度法). 设 $f(x)$ 在 x_k 附近连续可微, d_k 为搜索方向向量, $g_k = \nabla f(x_k)$. 由泰勒展开式得

$$f(x_k + \alpha d_k) = f(x_k) + \alpha g_k^T d_k + o(\alpha), \ \ \alpha > 0.$$

那么目标函数 $f(x)$ 在 x_k 处沿方向 d_k 下降的变化率为

$$\begin{aligned}\lim_{\alpha\to 0}\frac{f(x_k+\alpha d_k)-f(x_k)}{\alpha} &= \lim_{\alpha\to 0}\frac{\alpha g_k^T d_k + o(\alpha)}{\alpha}\\ &= g_k^T d_k \ = \ \|g_k\|\|d_k\|\cos\bar{\theta}_k,\end{aligned}$$

其中 $\bar{\theta}_k$ 是 g_k 与 d_k 的夹角. 显然, 对于不同的方向 d_k, 函数变化率取决于它与 g_k 夹角的余弦值. 要使变化率最小, 只有 $\cos\bar{\theta}_k = -1$, 即 $\bar{\theta}_k = \pi$ 时才能达到, 亦即

d_k 应该取 (3.2) 中的负梯度方向, 这也是将负梯度方向叫作最速下降方向的由来. 下面给出最速下降法的具体计算步骤.

算法 7 (**最速下降法**)

步 0 选取初始点 $x_0 \in \mathbb{R}^n$, 容许误差 $0 \le \varepsilon \ll 1$. 令 $k := 1$.

步 1 计算 $g_k = \nabla f(x_k)$. 若 $\|g_k\| \le \varepsilon$, 停算, 输出 x_k 作为近似最优解.

步 2 取方向 $d_k = -g_k$.

步 3 由线搜索技术确定步长因子 α_k.

步 4 令 $x_{k+1} := x_k + \alpha_k d_k$, $k := k+1$, 转步 1.

说明 步 3 中步长因子 α_k 的确定即可使用精确线搜索方法, 也可以使用非精确线搜索方法, 在理论上都能保证其全局收敛性. 若采用精确线搜索方法, 即

$$f(x_k + \alpha_k d_k) = \min_{\alpha \ge 0} f(x_k + \alpha d_k),$$

那么 α_k 应满足

$$\phi'(\alpha) = \frac{\mathrm{d}}{\mathrm{d}\alpha} f(x_k + \alpha d_k)|_{\alpha=\alpha_k} = \nabla f(x_k + \alpha_k d_k)^T d_k = 0.$$

由 (3.2) 有

$$\nabla f(x_{k+1})^T \nabla f(x_k) = 0,$$

即新点 x_{k+1} 处的梯度与旧点 x_k 处的梯度是正交的, 也就是说迭代点列所走的路线是锯齿型的, 故其收敛速度是很缓慢的 (至多线性收敛速度).

由 $d_k = -g_k$ 及 (2.16), 即

$$cos\theta_k = \frac{-g_k^T d_k}{\|g_k\|\|d_k\|} = \frac{-g_k^T(-g_k)}{\|g_k\|\|-g_k\|} = 1, \Rightarrow \theta_k = 0,$$

故条件 (2.15) 必然满足 $(0 \le \theta_k \le \frac{\pi}{2} - \mu,\ \mu > 0)$. 从而直接应用定理 11 和定理 12 即得到最速下降法的全局收敛性定理:

定理 13 设目标函数 $f(x)$ 连续可微且其梯度函数 $\nabla f(x)$ 是 Lipschitz 连续的, $\{x_k\}$ 由最速下降法产生, 其中步长因子 α_k 由精确线搜索, 或由 Wolfe 准则, 或由 Armijo 准则产生. 则有

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

下面的定理给出了最速下降法求解严格凸二次函数极小值问题时的收敛速度估计, 其证明可参阅有关文献, 此处省略不证.

定理 14 设矩阵 $H \in \mathbb{R}^{n\times n}$ 对称正定, $c \in \mathbb{R}^n$. 记 λ_1 和 λ_n 分别是 H 最大和最小特征值, $\kappa = \lambda_1/\lambda_n$. 考虑如下极小化问题

$$\min f(x) = c^T x + \frac{1}{2}x^T Hx.$$

设 $\{x_k\}$ 是用精确线搜索的最速下降法求解上述问题所产生的迭代序列, 则对于所有的 k, 下面的不等式成立

$$\|x_{k+1} - x^*\|_H \le \left(\frac{\kappa - 1}{\kappa + 1}\right)\|x_k - x^*\|_H, \tag{3.3}$$

其中, x^* 是问题的唯一解, $\|x\|_H = \sqrt{x^T Hx}$.

由上面的定理可以看出, 若条件数 κ 接近于 1 (即 H 的最大特征值和最小特征值接近时), 最速下降法是收敛很快的. 但当条件数 κ 较大时 (即 H 近似于病态时), 算法的收敛速度是很缓慢的.

下面我们给出基于 Armijo 非精确线搜索的最速下降法 Matlab 程序.

程序 4 (**最速下降法程序**)

```
function [x,val,k]=grad(fun,gfun,x0)
%功能: 用最速下降法求解无约束问题:  min f(x)
%输入:  x0是初始点, fun, gfun分别是目标函数和梯度
%输出:  x, val分别是近似最优点和最优值,  k是迭代次数.
maxk=5000;   %最大迭代次数
rho=0.5;sigma=0.4;
k=0;  epsilon=1e-5;
while(k<maxk)
    g=feval(gfun,x0);  %计算梯度
    d=-g;    %计算搜索方向
    if(norm(d)<epsilon), break; end
    m=0; mk=0;
    while(m<20)   %Armijo搜索
        if(feval(fun,x0+rho^m*d)<feval(fun,x0)+sigma*rho^m*g'*d)
            mk=m; break;
        end
```

```
            m=m+1;
        end
        x0=x0+rho^mk*d;
        k=k+1;
    end
    x=x0;
    val=feval(fun,x0);
```

例 11 利用程序 4 求解无约束优化问题

$$\min_{x\in\mathbb{R}^2} f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2.$$

该问题有精确解 $x^* = (1,1)^T$, $f(x^*) = 0$.

解 首先建立两个分别计算目标函数和梯度的 m 文件:

```
function f=fun(x)
f=100*(x(1)^2-x(2))^2+(x(1)-1)^2;
function g=gfun(x)
g=[400*x(1)*(x(1)^2-x(2))+2*(x(1)-1), -200*(x(1)^2-x(2))]';
```

我们利用程序 4, 终止准则取为 $\|\nabla f(x_k)\| \le 10^{-5}$. 取不同的初始点, 数值结果如下表.

表 4.1 最速下降法的数值结果.

初始点 (x_0)	迭代次数 (k)	目标函数值 $(f(x_k))$
$(0.0, 0.0)^T$	1159	1.1630×10^{-10}
$(2.0, 1.0)^T$	611	1.1416×10^{-10}
$(1.0, -1.0)^T$	1551	1.2251×10^{-10}
$(-1.0, -1.0)^T$	1499	9.2536×10^{-11}
$(-1.2, 1)^T$	1435	1.1985×10^{-10}
$(10, -10)^T$	1024	1.0156×10^{-10}

由上表可以看出, 最速下降法的收敛速度是比较缓慢的.

说明 上述程序的调用方式是:

```
x0=[-1.2 1]';
[x,val,k]=grad('fun','gfun',x0)
```

其中 fun, gfun 分别是求目标函数值及其梯度的 M 函数文件.

3.2　牛顿法及其 Matlab 实现

跟最速下降法一样, 牛顿法也是求解无约束优化问题最早使用的经典算法之一. 其基本思想是用迭代点 x_k 处的一阶导数 (梯度) 和二阶导数 (Hesse 阵) 对目标函数进行二次函数近似, 然后把二次模型的极小点作为新的迭代点, 并不断重复这一过程, 直至求得满足精度的近似极小点.

下面来推导牛顿法的迭代公式. 设 $f(x)$ 的 Hesse 阵 $G(x) = \nabla^2 f(x)$ 连续, 截取其在 x_k 处的泰勒展开式的前三项得

$$q_k(x) = f_k + g_k^T(x - x_k) + \frac{1}{2}(x - x_k)^T G_k(x - x_k),$$

其中 $f_k = f(x_k)$, $g_k = \nabla f(x_k)$, $G_k = \nabla^2 f(x_k)$. 求二次函数 $q_k(x)$ 的稳定点, 得

$$\nabla q_k(x) = g_k + G_k(x - x_k) = 0.$$

若 G_k 非奇异, 那么解上面的线性方程组 (记其解为 x_{k+1}) 即得牛顿法的迭代公式

$$x_{k+1} = x_k - G_k^{-1} g_k. \tag{3.4}$$

在迭代公式 (3.4) 中每步迭代需要求 Hesse 阵的逆 G_k^{-1}, 在实际计算中可通过 先解 $G_k d = -g_k$ 得 d_k, 然后令 $x_{k+1} = x_k + d_k$ 来避免求逆. 这样, 可以写出基本牛顿法的步骤如下:

算法 8 (**基本牛顿法**)

步 0　给定终止误差值 $0 \le \varepsilon \ll 1$, 初始点 $x_0 \in \mathbb{R}^n$. 令 $k : 0$.

步 1　计算 $g_k = \nabla f(x_k)$. 若 $\|g_k\| \le \varepsilon$, 停算, 输出 $x^* \approx x_k$.

步 2　计算 $G_k = \nabla^2 f(x_k)$, 并求解线性方程组得解 d_k:

$$G_k d = -g_k.$$

步 3　令 $x_{k+1} = x_k + d_k$, $k := k + 1$, 转步 1.

牛顿法最突出的优点是收敛速度快，具有局部二阶收敛性. 下面的定理表明了这一性质.

定理 15　设函数 $f(x)$ 有二阶连续偏导数, 在局部极小点 x^* 处, $G(x^*) = \nabla^2 f(x^*)$ 是正定的且 $G(x)$ 在 x^* 的一个邻域内是 Lipschitz 连续的. 如果初始点 x_0 充分靠近 x^*, 那么对一切 k, 牛顿迭代公式 (3.4) 是适定的, 并当 $\{x_k\}$ 为无穷点列时, 其极限为 x^*, 且收敛阶至少是二阶的.

证　由 $G(x^*)$ 的正定性及 f 二次连续可微可知, 存在 x^* 的一个邻域 $U(x^*)$, 使得对任意的 $x \in U(x^*)$, 都有 $G(x)$ 是一致正定的. 特别地, $\|G(x)^{-1}\|$ 在 $U_1(x^*)$ 上有界, 即存在常数 $M \geq 0$, 使得 $\|G(x)^{-1}\| \leq M, \forall\, x \in U_1(x^*)$. 又由 $G(x)$ 的连续性可知, 存在邻域 $U(x^*)$, 使得

$$\|G(x) - G(x^*)\| \leq \frac{1}{4M}, \quad \forall\, x \in U(x^*) \subseteq U_1(x^*).$$

因此, 当 $x_0 \in U(x^*)$ 时, 有

$$\begin{aligned}
\|x_1 - x^*\| &= \|x_0 - x^* - G_0^{-1} g_0\| \\
&\leq \|G_0^{-1}\| \|g(x_0) - g(x^*) - G_0(x_0 - x^*)\| \\
&\leq \|G_0^{-1}\| \left\| \int_0^1 G(x^* + \tau(x - x^*))(x - x^*)\mathrm{d}\tau - G_0(x_0 - x^*) \right\| \\
&\leq M \int_0^1 \|G(x^* + \tau(x_0 - x^*)) - G(x_0)\| \|x_0 - x^*\| \mathrm{d}\tau \\
&\leq M \left(\int_0^1 \|G(x^* + \tau(x_0 - x^*)) - G(x^*)\| \mathrm{d}\tau \right. \\
&\qquad \left. + \int_0^1 \|G(x_0) - G(x^*)\| \mathrm{d}\tau \right) \|x_0 - x^*\| \\
&\leq \frac{1}{2} \|x_0 - x^*\|. \qquad (3.5)
\end{aligned}$$

上式特别说明, $x_1 \in U(x^*)$. 类似地, 利用归纳法原理, 可证明对于所有的 $k \geq 1$ 有

$$\|x_{k+1} - x^*\| \leq \frac{1}{2} \|x_k - x^*\|.$$

因此 $\{x_k\} \subset U(x^*)$, 且 $x_k \to x^*$ $(k \to \infty)$. 进一步, 类似于 (3.5) 的推导, 可得

$$\begin{aligned}
\|x_{k+1} - x^*\| &\leq M \int_0^1 \|G(x^* + \tau(x_k - x^*)) - G(x_k)\| \|x_k - x^*\| \mathrm{d}\tau \\
&= o(\|x_k - x^*\|),
\end{aligned}$$

即 $\{x_k\}$ 超线性收敛于 x^*. 若 $G(x)$ 在 x^* 的一个邻域内 Lipschitz 连续, 则由上式

得

$$
\begin{aligned}
\|x_{k+1}-x^*\| &\le M\Big(\int_0^1 \|G(x^*+\tau(x_k-x^*))-G(x^*)\|\mathrm{d}\tau \\
&\qquad + \int_0^1 \|G(x^*)-G(x_k)\|\mathrm{d}\tau\Big)\|x_k-x^*\| \\
&\le LM\Big(\int_0^1 \tau\mathrm{d}\tau+1\Big)\|x_k-x^*\|^2 \\
&= \frac{3}{2}LM\|x_k-x^*\|^2,
\end{aligned}
$$

即 $\{x_k\}$ 二次收敛于 x^*. 证毕. □

上述定理指出, 初始点需要足够"靠近"极小点, 否则有可能导致算法不收敛. 由于实际问题的精确极小点一般是不知道的, 因此初始点的选取给算法的实际操作带来了很大的困难. 为了克服这一困难, 可引入线搜索技术以得到大范围收敛的算法, 即所谓的阻尼牛顿法. 我们给出一个基于 Armijo 搜索的阻尼牛顿法如下:

算法 9 (阻尼牛顿法)

步 0 给定终止误差值 $0 \le \varepsilon \ll 1$, $\delta \in (0,1)$, $\sigma \in (0,0.5)$. 初始点 $x_0 \in \mathbb{R}^n$. 令 $k:0$.

步 1 计算 $g_k = \nabla f(x_k)$. 若 $\|g_k\| \le \varepsilon$, 停算, 输出 $x^* \approx x_k$.

步 2 计算 $G_k = \nabla^2 f(x_k)$, 并求解线性方程组得解 d_k:

$$G_k d = -g_k \tag{3.6}$$

步 3 记 m_k 是满足下列不等式的最小非负整数 m:

$$f(x_k+\delta^m d_k) \le f(x_k) + \sigma\delta^m g_k^T d_k. \tag{3.7}$$

步 4 令 $x_{k+1} = x_k + \alpha_k d_k$, $k := k+1$, 转步 1.

我们给出算法 9 的全局收敛性定理如下:

定理 16 设函数 $f(x)$ 二次连续可微且存在常数 $\gamma > 0$, 使得

$$d^T\nabla^2 f(x)d \ge \gamma\|d\|^2,\ \forall\, d \in \mathbb{R}^n, x \in L(x_0), \tag{3.8}$$

其中 $L(x_0) = \{x | f(x) \le f(x_0)\}$. 设 $\{x_k\}$ 是由算法 9 产生的无穷点列, 则该点列收敛于 f 在水平集 $L(x_0)$ 中的唯一全局极小点.

证 由条件 (3.8) 知, f 是水平集 $L(x_0)$ 上的一致凸函数, 因此一定存在唯一的全局极小点 x^*, 且 x^* 是 $\nabla f(x)=0$ 的唯一解.

由于 f 是凸函数, 故水平集 $L(x_0)$ 是一个有界闭凸集. 注意到算法 9 的步 3, 序列 $\{f(x_k)\}$ 是单调下降的. 故显然有 $\{x_k\}\subset L(x_0)$.

由 (3.6) 和 (3.8) 不难得到

$$\|g_k\|\le \beta\|d_k\|,$$

其中 $\beta>0$ 是某个常数. 设 $\bar{x}$ 是 $\{x_k\}$ 的任意一个极限点, 则有 $f_k\to f(\bar{x})$. 记 θ_k 是负梯度方向 $-g_k$ 与牛顿方向 $d_k=-G_k^{-1}g_k$ 的夹角. 由上式和 (3.6), (3.8) 可推得

$$\begin{aligned}\cos\theta_k &= \frac{-g_k^Td_k}{\|g_k\|\|d_k\|}=\frac{d_k^TG_kd_k}{\|g_k\|\|d_k\|}\ge\frac{\gamma\|d_k\|^2}{\|g_k\|\|d_k\|}\\ &= \frac{\gamma\|d_k\|}{\|g_k\|}\ \ge\ \frac{\gamma}{\beta}\ >\ 0.\end{aligned}$$

则由定理 11 可得

$$\lim_{k\to\infty} g_k=0,$$

即 $\{x_k\}$ 的极限点都是稳定点. 由 f 的凸性知, 稳定点亦即 (全局) 极小点. 故由极小点的唯一性知, $\{x_k\}$ 收敛于 f 在水平集 $L(x_0)$ 上的全局极小点. 证毕. □

为了分析算法 9 的收敛速度, 需要用到下面的引理, 其详细的证明过程可参见文献 [4].

引理 3 *设函数 $f:\mathbb{R}^n\to\mathbb{R}$ 二次连续可微, 点列 $\{x_k\}$ 由算法 9 产生. 设 $\{x_k\}\to x^*$ 且 $g(x^*)=0$, $G(x^*)$ 正定. 那么, 若*

$$\lim_{k\to\infty}\frac{\|G(x_k)d_k+g_k\|}{\|d_k\|}=0, \tag{3.9}$$

则 (1) 当 k 充分大时, 步长因子 $\alpha_k\equiv 1$. (2) 点列 $\{x_k\}$ 超线性收敛于 x^*.

定理 17 *设定理 16 的条件成立, 点列 $\{x_k\}$ 由算法 9 产生. 则 $\{x_k\}$ 超线性收敛于 f 的全局极小点 x^*. 此外, 若 Hesse 阵 $G(x)$ 在 x^* 处 Lipschitz 连续, 则收敛阶至少是二阶的.*

证 定理 16 已证明 $\{x_k\} \to x^*$ 且 $g(x^*) = \nabla f(x^*) = 0$. 又由算法 9 的步 2 显然有条件 (3.9) 成立. 故由引理 3(1) 可知对于充分大的 k, $\alpha_k = 1$ 满足算法中的线搜索式. 因此, 由定理 15 立即得到 $\{x_k\}$ 至少二阶收敛于 x^*. 证毕. □.

下面我们给出基于 Armijo 非精确线搜索的阻尼牛顿法 Matlab 程序.

程序 5 (**阻尼牛顿法程序**)

```
function [x,val,k]=dampnm(fun,gfun,Hess,x0)
%功能: 用阻尼牛顿法求解无约束问题:  min f(x)
%输入: x0是初始点, fun, gfun, Hess 分别是求
%       目标函数值,梯度,Hesse 阵的函数
%输出:  x, val分别是近似最优点和最优值,  k是迭代次数.
maxk=100;   %给出最大迭代次数
rho=0.55;sigma=0.4;
k=0;  epsilon=1e-5;
while(k<maxk)
    gk=feval(gfun,x0); %计算梯度
    Gk=feval(Hess,x0);  %计算Hesse阵
    dk=-Gk\gk; %解方程组Gk*dk=-gk, 计算搜索方向
    if(norm(gk)<epsilon), break; end  %检验终止准则
    m=0; mk=0;
    while(m<20)   % 用Armijo搜索求步长
        if(feval(fun,x0+rho^m*dk)<feval(fun,x0)+sigma*rho^m*gk'*dk)
            mk=m; break;
        end
        m=m+1;
    end
    x0=x0+rho^mk*dk;
    k=k+1;
end
x=x0;
val=feval(fun,x);
```

例 12 利用程序 5 求解无约束优化问题

$$\min_{x\in\mathbb{R}^2} f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2.$$

该问题有精确解 $x^* = (1,1)^T$, $f(x^*) = 0$.

解 除了例 11 中建立的两个计算目标函数和梯度的 m 文件之外, 还需建立求 Hesse 阵的 m 文件:

```
function He=Hess(x)
n=length(x);
He=zeros(n,n);
He=[1200*x(1)^2-400*x(2)+2, -400*x(1);
    -400*x(1),              200     ];
```

我们利用程序 5, 终止准则取为 $\|\nabla f(x_k)\| \le 10^{-5}$. 取不同的初始点, 数值结果如下表.

表 4.2 阻尼牛顿法的数值结果.

初始点 (x_0)	迭代次数 (k)	目标函数值 ($f(x_k)$)
$(0,0)^T$	13	9.6238×10^{-15}
$(0.5,0.5)^T$	11	3.5183×10^{-19}
$(2,2)^T$	14	1.6322×10^{-14}
$(-1,-1)^T$	20	3.6221×10^{-17}
$(1,10)^T$	1	4.9309×10^{-28}
$(10,10)^T$	47	3.3426×10^{-17}
$(20,20)^T$	73	3.0386×10^{-17}

由上表可以看出, 阻尼牛顿法的收敛速度是比较令人满意的.

说明 上述程序的调用方式是:

```
x0=[-1.2 1]';
[x,val,k]=dampnm('fun','gfun','Hess',x0)
```

其中 fun, gfun, Hess 分别是求目标函数值和它的梯度及其 Hesse 阵的 M 函数文件.

3.3 修正牛顿法及其 Matlab 实现

从上一节的分析可知, 牛顿法具有不低于二阶的收敛速度, 这是它的优点. 但该算法要求目标函数的 Hesse 阵 $G(x) = \nabla^2 f(x)$ 在每个迭代点 x_k 处是正定的, 否则难以保证牛顿方向 $d_k = -G_k^{-1} g_k$ 是 f 在 x_k 处的下降方向. 为克服这一缺陷,

可对牛顿法进行修正. 修正的途径之一是将牛顿法和最速下降法结合起来, 构造所谓的“牛顿-最速下降混合算法”, 其基本思想是: 当 $\nabla^2 f(x_k)$ 正定时, 采用牛顿方向作为搜索方向; 否则, 若 $\nabla^2 f(x_k)$ 奇异, 或者虽然非奇异但牛顿方向不是下降方向, 则采用负梯度方向作为搜索方向. 我们写出详细的计算步骤如下:

算法 10 (**牛顿-最速下降混合算法**)

步 0　给定初始点 $x_0 \in \mathbb{R}^n$, 终止误差 $0 \le \varepsilon \ll 1$. 令 $k := 1$.

步 1　计算 $g_k = \nabla f(x_k)$. 若 $\|g_k\| \le \varepsilon$, 停算, 输出 x_k 作为近似极小点.

步 2　计算 $G_k = \nabla^2 f(x_k)$. 解方程组

$$G_k d + g_k = 0. \tag{3.10}$$

若 (4.10) 有解 d_k 且满足 $g_k^T d_k < 0$, 转步 3; 否则, 令 $d_k = -g_k$ 转步 3.

步 3　由线搜索技术确定步长因子 α_k.

步 4　令 $x_{k+1} = x_k + \alpha_k d_k$, $k := k + 1$, 转步 1.

对于算法 10, 利用定理 11, 定理 12 以及引理 3 不难证明下面的收敛性定理.

定理 18　设对任意的 $x_0 \in \mathbb{R}^n$, 水平集 $L(x_0) = \{x \,|\, f(x) \le f(x_0)\}$ 有界, 且函数 f 在包含 $L(x_0)$ 的一个有界闭凸集上二次连续可微. $\{x_k\}$ 由采用精确线搜索, 或 Wolfe 准则, 或 Armijo 准则确定步长因子的算法 10 产生的迭代序列, 且存在 $\{x_k\}$ 的一个极限点 x^*, 使得 $G(x^*)$ 正定. 则有

$$\lim_{k\to\infty} \inf \|g_k\| = 0,$$

以及 $\{x_k\}$ 超线性收敛于 x^*. 进一步, 若 $G(x)$ 在 x^* 处是 Lipschitz 连续的, 则收敛速度至少是二阶的.

上述的修正牛顿法克服了牛顿法要求 Hesse 阵 $G(x_k) = \nabla^2 f(x_k)$ 正定的缺陷. 克服这一缺陷还有其它的方法和途径. 例如, 引进阻尼因子 $\mu_k \ge 0$, 即在每一迭代步适当地选取参数 μ_k 使得矩阵 $A_k = G(x_k) + \mu_k I$ 正定. 我们写出算法步骤如下:

算法 11 (**修正牛顿法**)

步 0　给定参数 $\delta \in (0,1)$, $\tau \in [0,1]$, $\sigma \in (0, 0.5)$, 终止误差 $0 \le \varepsilon \ll 1$. 初始点 $x_0 \in \mathbb{R}^n$. 令 $k := 0$.

步 1　计算 $g_k = \nabla f(x_k)$, $\mu_k = \|g_k\|^{1+\tau}$. 若 $\|g_k\| \le \varepsilon$, 停算, 输出 x_k 作为近似极小点.

步 2　计算 Hesee 阵 $G_k = \nabla^2 f(x_k)$. 解线性方程组

$$(G_k + \mu_k I)d = -g_k, \tag{3.11}$$

得解 d_k.

步 3　令 m_k 是满足下列不等式的最小非负整数 m:

$$f(x_k + \delta^m d_k) \le f(x_k) + \sigma\delta^m g_k^T d_k. \tag{3.12}$$

令 $\alpha_k = \delta^{m_k}$, $x_{k+1} := x_k + \alpha_k d_k$.

步 4　令 $k := k+1$, 转步 1.

下面的定理给出了算法 11 的全局收敛性, 其证明可参看文献 [4].

定理 19　设函数 $f: \mathbb{R}^n \to \mathbb{R}$ 有下界且二次连续可微, $G(x) = \nabla^2 f(x)$ 半正定且 Lipschitz 连续. 则由算法 11 产生的迭代序列 $\{x_k\}$ 的任何极限点都是问题 (3.1) 的解.

已有研究证明, 当目标函数的 Hesse 阵 $G(x) = \nabla^2 f(x)$ 在极小点 x^* 处奇异时, 牛顿法的收敛速度可能会降低为线性收敛速度. 下面我们给出奇异解的概念.

定义 14　若在问题 (3.1) 的极小点 x^* 处 Hesse 阵 $G(x^*)$ 奇异, 则称 x^* 是问题 (3.1) 的奇异解.

当问题 (3.1) 有奇异解时, 其解可能不唯一, 此时我们用 X 表示其解集, 即

$$X = \{x^* \mid f(x^*) = \min f(x),\ x \in \mathbb{R}^n\}.$$

定义 15　设 $x^* \in X$, 函数 $\theta: \mathbb{R}^n \to \mathbb{R}_+$. 若存在 x^*的邻域 $U(x^*)$ 及常数 $\gamma > 0$, 使得

$$\theta(x) \ge \gamma \mathrm{dist}(x, X), \quad x \in U(x^*), \tag{3.13}$$

其中 $\mathrm{dist}(x, X)$ 表示点 x 到集合 X 的距离, 则称函数 θ 在邻域 $U(x^*)$ 内对问题 (3.1) 解集合 X 提供了一个局部误差界.

下面的定理给出了算法 11 的局部收敛速度的估计, 其证明可参看文献 [4].

定理 20 设定理 19 的条件成立. 若由算法 11 产生的迭代序列 $\{x_k\}$ 有子列 $\{x_k : k \in K\}$ 收敛于 $x^* \in X$, 且函数 $\|g(x)\|$ 在 x^* 的某邻域内对问题 (3.1) 提供了一个局部误差界, 则当 $k \in K$ 充分大时, $\alpha_k \equiv 1$, 且子列 $\{x_k : k \in K\}$ 二阶收敛于 x^*, 即存在常数 $\beta > 0$ 使得

$$\mathrm{dist}(x_{k+1}, X) \leq \beta \mathrm{dist}(x_k, X).$$

下面我们给出算法 11 (修正牛顿法) 的 Matlab 程序.

程序 6 (**修正牛顿法 Matlab 程序**)

```
function [x,val,k]=revisenm(fun,gfun,Hess,x0)
% 功能: 用修正牛顿法求解无约束问题:  min f(x)
%输入:  x0是初始点, fun, gfun, Hess 分别是求
%        目标函数值,梯度,Hesse 阵的函数
%输出:  x, val分别是近似最优点和最优值,  k是迭代次数.
n=length(x0); maxk=150;
rho=0.55;sigma=0.4; tau=0.0;
k=0;  epsilon=1e-5;
while(k<maxk)
    gk=feval(gfun,x0); % 计算梯度
    muk=norm(gk)^(1+tau);
    Gk=feval(Hess,x0);  % 计算Hesse阵
    Ak=Gk+muk*eye(n);
    dk=-Ak\gk; %解方程组Gk*dk=-gk, 计算搜索方向
    if(norm(gk)<epsilon), break; end  %检验终止准则
    m=0; mk=0;
    while(m<20)   %用Armijo搜索求步长
        if(feval(fun,x0+rho^m*dk)<feval(fun,x0)+sigma*rho^m*gk'*dk)
            mk=m; break;
        end
        m=m+1;
    end
    x0=x0+rho^mk*dk;
    k=k+1;
end
x=x0;
val=feval(fun,x);
```

例 13 利用程序 6 求解无约束优化问题

$$\min_{x\in\mathbb{R}^2} f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2.$$

该问题有精确解 $x^* = (1,1)^T$, $f(x^*) = 0$.

解 前面例 11 和例 12 中已经建立的计算目标函数及其梯度和 Hesee 阵的 m 文件, 可以重新利用. 取终止准则值为 $\|\nabla f(x_k)\| \le 10^{-5}$, 利用程序 6, 取不同的初始点, 数值结果如下表.

表 4.3 修正牛顿法的数值结果.

初始点 (x_0)	迭代次数 (k)	目标函数值 ($f(x_k)$)
$(0,0)^T$	16	4.7808×10^{-12}
$(0.5,0.5)^T$	10	2.4524×10^{-15}
$(2,2)^T$	17	1.3250×10^{-19}
$(-1,-1)^T$	23	2.3697×10^{-12}
$(1,10)^T$	1	4.9309×10^{-28}
$(10,10)^T$	46	4.5469×10^{-18}
$(20,20)^T$	76	9.1654×10^{-13}

由上表可以看出, 修正牛顿法的收敛速度不如阻尼牛顿法快, 因为矩阵 $A_k = G_k + \mu_k I$ 只是 Hesee 阵 G_k 的一个近似.

说明 上述程序的调用方式是:

```
x0=[-1.2 1]';
[x,val,k]=revisenm('fun','gfun','Hess',x0)
```

其中 fun, gfun, Hess 分别是求目标函数值和它的梯度及其 Hesse 阵的 M 函数文件.

习 题 3

1. 用最速下降法求 $f(x_1,x_2) = 3x_1^2 + 2x_2^2 - 4x_1 - 6x_2$ 的极小值.
2. 分别用牛顿法和阻尼牛顿法求函数 $f(x_1,x_2) = 4x_1^2 + x_2^2 - 8x_1 - 4x_2$ 的极小点.
3. 用最速下降法程序求函数 $f(x_1,x_2) = (x_1-2)^4 + (x_1-2x_2)^2$ 的极小值点, 取初始点 $x_0 = (0,3)^T$.

4. 用牛顿法程序求 Rosenbrock 函数 $f(x)=100(x_2-x_1^2)^2+(x_1-1)^2$ 的极小点, 取初始点 $x_0=(-1,2,1)^T$.

5. 设二次函数为 $f(x)=\frac{1}{2}x^THx+b^Tx$, 其中 H 是 n 阶对称正定阵. 证明最速下降法求 $f(x)$ 的极小点时, 序列 $\{x_k\}$ 由 $x_{k+1}=x_k-\frac{g_k^Tg_k}{g_k^THg_k}g_k,\ k=0,1,2,\cdots$ 确立, 其中 x_0 为给定的初始点, $g_k=Hx_k+b$.

6. 设目标函数 $f(x)=4x_1^2+x_2^2-x_1^2x_2$, 记 $x^a=(1,1)^T, x^b=(3,4)^T, x^c=(2,0)^T$.

(1) 分别取初始点 $x_0^a=x_a^a, x_0^b=x_b^b$,用标准牛顿法计算前三次迭代;

(2) 注意序列 $\{x_k^a\}$ 和 $\{x_k^b\}$ 分别收敛到 $x_*^a=(0,0)^T$ 和 $x_*^b=(2\sqrt{2},4)^T$, 证明 x_*^a 是 $f(x)$ 的极小点, 即 x_*^b 是 f 的一个鞍点;

(3) 取 $x_0^c=x_c^c$, 用牛顿法计算 x_1^c, 这将给出牛顿法失败的一种情形.

7. 给定函数 $f(x)=(6+x_1+x_2)^2+(2-3x_1-3x_2-x_1x_2)^2$, 求在点 $\bar{x}=(-4,6)^T$ 处的最速下降方向和牛顿方向.

8. 考虑函数 $f(x)=x_1^2+4x_2^2-4x_1-8x_2$.

(1) 证明: 若从 $x_0=(0,0)^T$ 出发, 用最速下降法求极小点 $\bar{x}$, 则不能经有限步迭代达到 $\bar{x}$.

(2) 是否存在 x_0, 使得从 x_0 出发, 用最速下降法求 $f(x)$ 的极小点, 经有限步迭代即收敛?

9. 设函数 $f(x)=\frac{1}{2}x^THx+b^Tx$, 其中 H 对称正定. 又设 $x_0(\neq\bar{x})$ 可表示为 $x_0=\bar{x}+\mu d$, 其中 $\bar{x}$ 是 $f(x)$ 的极小点, d 是 H 的属于特征值 λ 的特征向量. 证明:

(1) $\nabla f(x_0)=\mu\lambda d$;

(2) 如果从 x_0 出发, 沿最速下降作精确的一维搜索，则一步迭代达到极小点 $\bar{x}$.

第四章　共轭梯度法

前面介绍的最速下降法和牛顿法都具有其自身的局限性. 本章将要介绍的共轭梯度法是介于最速下降法与牛顿法之间的一种无约束优化算法, 它具有超线性收敛速度, 而且算法结构简单, 容易编程实现. 此外, 跟最速下降法相类似, 共轭梯度法只用到了目标函数及其梯度值, 避免了二阶导数 (Hesse 阵) 的计算, 从而降低了计算量和存储量, 因此它是求解无约束优化问题的一种比较有效而实用的算法.

4.1　共轭方向法

共轭方向法的基本思想是在求解 n 维正定二次目标函数极小点时产生一组共轭方向作为搜索方向, 在精确线搜索条件下算法至多迭代 n 步即能求得极小点. 经过适当的修正后共轭方向法可以推广到求解一般非二次目标函数情形. 下面先介绍共轭方向的概念.

定义 16　设 G 是 n 阶对称正定矩阵, 若 n 维向量组 $d_1, d_2, \cdots, d_m\,(m \le n)$ 满足

$$d_i^T G d_j = 0, \quad i \ne j,$$

则称 $d_1, d_2, \cdots, d_m$ 是 G 共轭的.

显然, 向量组的共轭是正交的推广, 即当 $G = I$(单位阵) 时, 上述定义变成向量组正交的定义. 此外, 不难证明, 对称正定矩阵 G 的共轭向量组必然是线性无关的.

下面我们考虑求解正定二次目标函数极小点的共轭方向法. 设

$$\min f(x) = \frac{1}{2}x^T G x + b^T x + c, \tag{4.1}$$

其中 G 是 n 阶对称正定阵, b 为 n 维常向量, c 为常数. 我们有下面的算法:

算法 12 (共轭方向法)

步 0 给定迭代精度 $0 \le \varepsilon \ll 1$ 和初始点 x_0. 计算 $g_0 = \nabla f(x_0)$. 选取初始方向 d_0 使得 $d_0^T g_0 < 0$. 令 $k := 0$.

步 1 若 $\|g_k\| \le \varepsilon$, 停算, 输出 $x^* \approx x_k$.

步 2 利用精确线搜索方法确定搜索步长 α_k.

步 3 令 $x_{k+1} := x_k + \alpha_k d_k$, 并计算 $g_{k+1} = \nabla f(x_{k+1})$.

步 4 选取 d_{k+1} 满足下降性和共轭性条件:

$$d_{k+1}^T g_{k+1} < 0, \quad d_{k+1}^T G d_i = 0, \quad i = 0, 1, \cdots, k.$$

步 5 令 $k := k + 1$, 转步 1.

下面给出算法 12 的收敛性定理.

定理 21 设目标函数 f 由 (4.1) 定义. $\{x_k\}$ 是算法 12 产生的迭代序列. 则每一步迭代点 x_{k+1} 都是 $f(x)$ 在 x_0 和方向 $d_0, d_1, \cdots, d_k$ 所张成的线性流形

$$S_k = \left\{ x \,\Big|\, x = x_0 + \sum_{i=0}^{k} \alpha_i d_i, \ \forall \alpha_i \right\}$$

中的极小点. 特别, $x_n = x^* = -G^{-1}b$ 是问题 (4.1) 的唯一极小点.

证 由算法 12 可知, $d_0, d_1, \cdots, d_{n-1}$ 是 G 共轭的, 因而是线性无关的, 故有 $S_{n-1} = \mathbb{R}^n$. 于是我们只需证明 x_{k+1} 是 f 在线性流形 S_k 中的极小点即可.

显然有

$$x_{k+1} = x_k + \alpha_k d_k = \cdots = x_0 + \sum_{i=0}^{k} \alpha_i d_i \in S_k.$$

另一方面, 对任何 $x \in S_k$, 存在 $\beta_i \in \mathbb{R}$, $i = 0, 1, \cdots, k$, 使得

$$x = x_0 + \sum_{i=0}^{k} \beta_i d_i.$$

记

$$h_{k+1} = x - x_{k+1} = \sum_{i=0}^{k} (\beta_i - \alpha_i) d_i.$$

利用泰勒展开公式, 有

$$\begin{aligned} f(x) &= f_{k+1} + g_{k+1}^T h_{k+1} + \frac{1}{2} h_{k+1}^T G h_{k+1} \\ &\geq f_{k+1} + g_{k+1}^T h_{k+1} \\ &= f_{k+1} + \sum_{i=0}^{k} (\beta_i - \alpha_i) g_{k+1}^T d_i. \end{aligned}$$

下面只需证明

$$g_{k+1}^T d_i, \quad \forall\, i = 0, 1, \cdots, k \tag{4.2}$$

即可. 事实上, 因

$$g_{j+1} - g_j = G(x_{j+1} - x_j) = \alpha_j G d_j,$$

故当 $i < k$ 时有

$$\begin{aligned} g_{k+1}^T d_i &= g_{i+1}^T d_i + \sum_{j=i+1}^{k} (g_{j+1} - g_j)^T d_i \\ &= g_{i+1}^T d_i + \sum_{j=i+1}^{k} \alpha_j d_j^T G d_i = 0, \end{aligned}$$

其中上式的第一项与求和项为 0 分别由精确线搜索和共轭性得到. 当 $i = k$ 直接由精确线搜索可得 $g_{k+1}^T d_k = 0$. 从而 (4.2) 式成立. 至此, 定理的结论已经得到证明. □

注 从定理 21 可知, 在精确线搜索下, 用算法 12 求解正定二次目标函数极小化问题 (4.1), 至多在 n 步内即可求得其唯一的极小点. 这种能在有限步内求得二次函数极小点的性质通常称为二次终止性.

4.2　共轭梯度法

共轭梯度法是在每一迭代步利用当前点处的最速下降方向来生成关于凸二次函数 f 的 Hesee 阵 G 的共轭方向, 并建立求 f 在 $\mathbb{R}^n$ 上的极小点的方法. 这一方法最早是由 Hesteness 和 Stiefel 于 1952 年为求解对称正定线性方程组而提出来的, 后经 Fletcher 等人研究并应用于无约束优化问题取得了丰富的成果, 共轭梯度法也因此成为当前求解无约束优化问题的重要算法类.

设函数 f 由 (4.1) 所定义, 则 f 的梯度和 Hesse 阵分别为

$$g(x) = \nabla f(x) = Gx + b, \quad G(x) = \nabla^2 f(x) = G. \tag{4.3}$$

下面我们来讨论算法 12 中共轭方向的构造. 我们取初始方向 d_0 为初始点 x_0 处的负梯度方向, 即

$$d_0 = -\nabla f(x_0) = -g_0. \tag{4.4}$$

从 x_0 出发沿 d_0 方向进行精确线搜索得求步长 α_0, 令

$$x_1 = x_0 + \alpha_0 d_0,$$

其中 α_0 满足条件

$$\nabla f(x_1)^T d_0 = g_1^T d_0 = 0. \tag{4.5}$$

在 x_1 处, 用 f 在 x_1 的负梯度方向 $-g_1$ 与 d_0 的组合来生成 d_1, 即

$$d_1 = -g_1 + \beta_0 d_0, \tag{4.6}$$

然后选取系数 β_0 使 d_1 与 d_0 关于 G 共轭, 即令

$$d_1^T G d_0 = 0 \tag{4.7}$$

来确定 β_0. 将 (4.6) 代入 (4.7) 得

$$\beta_0 = \frac{g_1^T G d_0}{d_0^T G d_0}. \tag{4.8}$$

由 (4.3) 得

$$g_1 - g_0 = G(x_1 - x_0) = \alpha_0 G d_0. \tag{4.9}$$

另外, 由定理 21 可知 $g_2^T d_i = 0\,(i = 0, 1)$. 故由 (4.4)~(4.6) 可得

$$g_2^T g_0 = 0, \;\; g_2^T g_1 = 0, \;\; d_0^T g_0 = -g_0^T g_0, \;\; d_1^T g_1 = -g_1^T g_1.$$

现假设已得到相互共轭的搜索方向 $d_0, d_1, \cdots, d_{k-1}$, 精确线搜索得到的步长为 $\alpha_0, \alpha_1, \cdots, \alpha_{k-1}$, 且满足

$$\begin{cases} d_{k-1}^T G d_i = 0, \quad i = 0, 1, \cdots, k-2, \\ d_i^T g_i = -g_i^T g_i, \quad i = 0, 1, \cdots, k-1, \\ g_k^T g_i = 0, \;\; g_k^T d_i = 0, \quad i = 0, 1, \cdots, k-1. \end{cases} \tag{4.10}$$

现令

$$d_k = -g_k + \beta_{k-1} d_{k-1} + \sum_{i=0}^{k-2} \beta_k^{(i)} d_i, \tag{4.11}$$

其中 β_{k-1}, $\beta_k^{(i)}\,(i=0,1,\cdots,k-2)$ 的选择要满足

$$d_k^T G d_i = 0, \;\; i = 0,1,\cdots,k-1. \tag{4.12}$$

用 $d_i^T G\,(i=0,1,\cdots,k-1)$ 左乘 (4.11) 得

$$\beta_{k-1} = \frac{g_k^T G d_{k-1}}{d_{k-1}^T G d_{k-1}}, \;\; \beta_k^{(i)} = \frac{g_k^T G d_i}{d_i^T G d_i}, \;\; i = 0,1,\cdots,k-2. \tag{4.13}$$

类似于 (4.9), 我们有

$$g_{i+1} - g_i = G(x_{i+1} - x_i) = \alpha_i G d_i, \;\; i = 0,1,\cdots,k-1,$$

及

$$\alpha_i G d_i = g_{i+1} - g_i, \;\; i = 0,1,\cdots,k-1. \tag{4.14}$$

于是由归纳法假设 (4.10) 可得

$$\beta_k^{(i)} = \frac{g_k^T G d_i}{d_i^T G d_i} = \frac{g_k^T (g_{i+1} - g_i)}{d_i^T (g_{i+1} - g_i)} = 0, \;\; i = 0,1,\cdots,k-2.$$

于是, 第 k 步的搜索方向为

$$d_k = -g_k + \beta_{k-1} d_{k-1}, \tag{4.15}$$

其中 β_{k-1} 由 (4.13) 确定, 即

$$\beta_{k-1} = \frac{g_k^T G d_{k-1}}{d_{k-1}^T G d_{k-1}}. \tag{4.16}$$

同时有 $d_k^T g_k = -g_k^T g_k$. 这样 (4.4), (4.15) 和 (4.16) 确定了一组由负梯度方向形成的共轭方向, 而把沿着这组方向进行迭代的方法称之为共轭梯度法.

上面得推导实际上已经证明了下述结论:

定理 22 对于正定二次函数的极小化问题 (4.1), 由 (4.4), (4.15) 和 (4.16) 确定搜索方向 d_k 并采用精确线搜索确定步长因子 α_k 的共轭方向法, 至多 n 步迭代即可求得问题 (4.1) 的极小点, 并且对所有的 $k\,(1 \le k \le n)$, 有

$$\begin{aligned} &d_k^T G d_i = 0, \;\; i = 0,1,\cdots,k-1, \\ &d_i^T g_i = -g_i^T g_i, \;\; i = 0,1,\cdots,k, \\ &g_{k+1}^T g_i = 0, \; g_{k+1}^T d_i = 0, \;\; i = 0,1,\cdots,k. \end{aligned}$$

为了使算法能够适应于求解非二次目标函数的极小点, 需要设法消去 (4.16) 中的矩阵 G. 由定理 22 及 (4.14) 得

$$\begin{aligned}\alpha_{k-1}g_k^TGd_{k-1} &= g_k^T(g_k-g_{k-1})=g_k^Tg_k,\\ \alpha_{k-1}d_{k-1}^TGd_{k-1} &= (-g_{k-1}+\beta_{k-2}d_{k-2})^T(g_k-g_{k-1})\\ &= g_{k-1}^Tg_{k-1}.\end{aligned}$$

由此, (4.16) 可化为

$$\beta_{k-1}=\frac{g_k^Tg_k}{g_{k-1}^Tg_{k-1}}. \tag{4.17}$$

下面我们给出共轭梯度法求解无约束优化问题 (4.1) 极小点的算法步骤.

算法 13 (**共轭梯度法**)

步 0　给定迭代精度 $0\le\varepsilon\ll 1$ 和初始点 x_0. 计算 $g_0=\nabla f(x_0)$. 令 $k:=0$.

步 1　若 $\|g_k\|\le\varepsilon$, 停算, 输出 $x^*\approx x_k$.

步 2　计算搜索方向 d_k :

$$d_k=\begin{cases}-g_k, & k=0,\\ -g_k+\beta_{k-1}d_{k-1}, & k\ge 1,\end{cases}$$

其中当 $k\ge 1$ 时, β_{k-1} 由 (4.17) 确定.

步 3　利用精确线搜索方法确定搜索步长 α_k.

步 4　令 $x_{k+1}:=x_k+\alpha_kd_k$, 并计算 $g_{k+1}=\nabla f(x_{k+1})$.

步 5　令 $k:=k+1$, 转步 1.

注 公式 (4.17) 是由 Fletcher 和 Reeves 给出的, 故称之为 Fletcher-Reeves 公式, 简称 FR 公式, 算法 13 也称之为 FR 共轭梯度法. 除 FR 公式外，尚有下列

著名公式:

$$\begin{aligned}
\beta_k &= \frac{g_{k+1}^T g_{k+1}}{-d_k^T g_k}, && \text{(Dixon 公式)},\\
\beta_k &= \frac{g_{k+1}^T g_{k+1}}{d_k^T(g_{k+1}-g_k)}, && \text{(Dai-Yuan 公式)},\\
\beta_k &= \frac{g_{k+1}^T (g_{k+1}-g_k)}{d_k^T(g_{k+1}-g_k)}, && \text{(Crowder-Wolfe 公式)},\\
\beta_k &= \frac{g_{k+1}^T (g_{k+1}-g_k)}{d_k^T(g_{k+1}-g_k)}, && \text{(Hesteness-Stiefel, HS 公式)},\\
\beta_k &= \frac{g_{k+1}^T (g_{k+1}-g_k)}{g_k^T g_k}, && \text{(Polak, Ribiére, Polyak, PRP 公式)}.
\end{aligned}$$

下面我们来证明算法 13 的收敛性定理.

定理 23 设 $\{x_k\}$ 是由算法 13 产生的序列, 假定函数 $f(x)$ 一阶连续可微且水平集 $\mathcal{L}(x_0)=\{x\,|\,f(x)\le f(x_0)\}$ 是有界的. 那么算法 13 或者有限步终止, 或者 $\lim\limits_{k\to\infty} g(x_k)=0$.

证 不失一般性, 不妨假设 $\{x_k\}$ 是无穷序列. 此时有 $g(x_k)\neq 0$. 因 $d_k=-g_k+\beta_{k-1}d_{k-1}$, 故有

$$g_k^T d_k = -\|g_k\|^2+\beta_{k-1}g_k^T d_{k-1} = -\|g_k\|^2<0,$$

即 d_k 是下降方向. 从而由精确线搜索规则可知, $\{f(x_k)\}$ 是单调下降的, 故 $\{x_k\}\subset\mathcal{L}(x_0)$. 于是 $\{x_k\}$ 是有界的, 因而必有聚点 x^*, 即存在子列 $\{x_k : k\in K_1\}$ 收敛到 x^*. 由 f 的连续性, 有

$$f^* = \lim_{k(\in K_1)\to\infty} f(x_k) = f\Big(\lim_{k(\in K_1)\to\infty} x_k\Big) = f(x^*).$$

类似地, $\{x_{k+1} : k\in K_1\}$ 也是有界序列, 故存在子列 $\{x_{k+1} : k\in K_2\}$ 收敛到 $\bar{x}^*$, 这里 $K_2\subset K_1$ 是无穷子序列. 于是可得

$$f^* = \lim_{k(\in K_2)\to\infty} f(x_k) = f\Big(\lim_{k(\in K_2)\to\infty} x_k\Big) = f(\bar{x}^*).$$

故有

$$f(\bar{x}^*) = f(x^*) = f^*. \tag{4.18}$$

下面用反证法证明 $g(x^*) = 0$. 如果不然, 即 $g(x^*) \neq 0$, 则对于充分小的 $\alpha > 0$, 有

$$f(x^* + \alpha d^*) < f(x^*).$$

注意到, 对任意的 $\alpha > 0$, 有

$$f(x_{k+1}) = f(x_k + \alpha_k d_k) \leq f(x_k + \alpha d_k).$$

对于 $k \in K_2 \subset K_1$, 令 $k \to \infty$ 对上式取极限得

$$f(\bar{x}^*) \leq f(x^* + \alpha d^*) < f(x^*),$$

这与 (4.18) 式矛盾, 从而证明了 $g(x^*) = 0$. 证毕. □

若在算法 13 中采用非精确搜索确定步长因子 α_k, 比如 Wolfe 准则 (2.10) 和 (2.11), 则利用一般下降类算法的全局收敛性定理, 可以得到非精确搜索下的共轭梯度法的收敛性定理.

定理 24 设 $\{x_k\}$ 是由算法 13 利用 Wolfe 准则 (2.10)-(2.11) 产生的序列, 假定函数 $f(x)$ 一阶连续可微且有下界, 其梯度函数 $g(x)$ 在 $\mathbb{R}^n$ 上 Lipschitz 连续, 即存在常数 $L > 0$, 使得

$$\|g(u) - g(v)\| \leq L\|u - v\|, \quad \forall\, u, v \in \mathbb{R}^n.$$

若选取的搜索方向 d_k 与 $-g_k$ 的夹角 θ_k 满足条件

$$0 \leq \theta_k \leq \frac{\pi}{2} - \mu, \quad \mu \in \left(0, \frac{\pi}{2}\right).$$

那么算法 13 或者有限步终止, 或者 $\lim\limits_{k\to\infty} g(x_k) = 0$.

证 不失一般性, 不妨假设 $\{x_k\}$ 是无穷序列. 由 Lipschitz及连续条件和 Wolfe 准则 (2.11) 式得

$$\begin{aligned}\alpha_k L\|d_k\| &\geq d_k^T[g(x_k + \alpha_k d_k) - g_k] \geq -(1-\sigma)d_k^T g_k \\ &= (1-\sigma)\|d_k\|\|g_k\|\cos\theta_k,\end{aligned}$$

即

$$\alpha_k\|d_k\| \geq \frac{(1-\sigma)}{L}\|g_k\|\cos\theta_k.$$

于是利用上式及 Wolfe 准则 (2.10) 式可得

$$\begin{aligned}
&f(x_k) - f(x_k + \alpha_k d_k) \\
&\geq -\rho\alpha_k d_k^T g_k = \rho\alpha_k \|d_k\| \|g_k\| \cos\theta_k \\
&\geq \rho\frac{(1-\sigma)}{L}\|g_k\|^2 \cos^2\theta_k \\
&\geq \frac{\rho(1-\sigma)}{L}\|g_k\|^2 \sin^2\mu.
\end{aligned}$$

注意到 $f(x)$ 是有下界的, 由上式不难推得

$$\sum_{k=0}^{\infty} \|g\|^2 < +\infty,$$

这蕴含了当 $k \to \infty$ 时, 有 $\|g_k\| \to 0$. 证毕. □

4.3　共轭梯度法的 Matlab 程序

在共轭梯度法的实际使用中, 通常在迭代 n 步或 $n+1$ 步之后, 重新取负梯度方向作为搜索方向, 我们称之为再开始共轭梯度法. 这是因为对于一般非二次函数而言, n 步迭代后共轭梯度法产生的搜索方向往往不再具有共轭性. 而对于大规模问题, 常常每 $m\,(m < n$ 或 $m \ll n)$ 步就进行再开始. 此外, 当搜索方向不是下降方向时, 也插入负梯度方向作为搜索方向.

本节给出基于 Armijo 非精确线搜索的再开始 FR 共轭梯度法的 Matlab 程序.

程序 7 (FR 共轭梯度法程序)

```
function [x,val,k]=frcg(fun,gfun,x0)
% 功能: 用FR共轭梯度法求解无约束问题:  min f(x)
%输入:  x0是初始点, fun, gfun分别是目标函数和梯度
%输出:  x, val分别是近似最优点和最优值,  k是迭代次数.
maxk=5000;   %最大迭代次数
rho=0.6;sigma=0.4;
k=0;  epsilon=1e-4;
n=length(x0);
while(k<maxk)
    g=feval(gfun,x0);  %计算梯度
```

```
        itern=k-(n+1)*floor(k/(n+1));
        itern=itern+1;
        %计算搜索方向
        if(itern==1)
            d=-g;
        else
            beta=(g'*g)/(g0'*g0);
            d=-g+beta*d0;  gd=g'*d;
            if(gd>=0.0)
                d=-g;
            end
        end
        if(norm(g)<epsilon), break; end  %检验终止条件
        m=0; mk=0;
        while(m<20)  %Armijo搜索
            if(feval(fun,x0+rho^m*d)<feval(fun,x0)+sigma*rho^m*g'*d)
                mk=m; break;
            end
            m=m+1;
        end
        x0=x0+rho^mk*d;
        val=feval(fun,x0);
        g0=g;  d0=d;
        k=k+1;
    end
    x=x0;
    val=feval(fun,x);
```

例 14 利用程序 7 求解无约束优化问题

$$\min_{x\in\mathbb{R}^2} f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2.$$

该问题有精确解 $x^* = (1,1)^T$, $f(x^*) = 0$.

解 利用程序 7, 终止准则取为 $\|\nabla f(x_k)\| \le 10^{-4}$. 取不同的初始点, 数值结果如下表.

表 5.1 FR 共轭梯度法的数值结果.

初始点 (x_0)	迭代次数 (k)	目标函数值 ($f(x_k)$)
$(0,0)^T$	122	7.2372×10^{-9}
$(0.5,0.5)^T$	44	3.5498×10^{-10}
$(1.2,-1)^T$	56	1.1698×10^{-8}
$(-1.2,1)^T$	44	2.9396×10^{-9}
$(-1.2,-1)^T$	58	2.0235×10^{-9}

由上表可以看出, 阻尼牛顿法的收敛速度是比较令人满意的.

说明　上述程序的调用方式是:

```
x0=[-1.2 1]';
[x,val,k]=frcg('fun','gfun',x0)
```

其中 fun, gfun 分别是求目标函数值及其梯度的 M 函数文件.

习 题 4

1. 证明向量 $\alpha_1=(1,0)^T$ 和 $\alpha_2=(3,-2)^T$ 关于矩阵

$$A=\begin{pmatrix}2&3\\3&5\end{pmatrix}$$

共轭.

2. 给定矩阵

$$A=\begin{pmatrix}1&2\\2&5\end{pmatrix},\quad B=\begin{pmatrix}1&-1&0\\-1&2&0\\0&0&3\end{pmatrix}.$$

试关于矩阵 A 和 B 各求出一组共轭方向.

3. 设 $f(x)=\dfrac{1}{2}x^THx+b^Tx$, 其中

$$H=\begin{pmatrix}4&2\\2&4\end{pmatrix},\quad b=\begin{pmatrix}3\\3\end{pmatrix}.$$

(1) 证明: $d_0=(1,0)^T$ 与 $d_1=(-1,2)^T$ 关于 H 共轭;

(2) 以 $x_0=(0,0)^T$ 为初始点, d_0 和 d_1 为搜索方向, 用精确线搜索求 f 的极小点.

4. 设 H 为 n 阶正定阵. $u_1, \cdots, u_n \in \mathbb{R}^n$ 线性无关, d_k 由下列方式产生:

$$d_1 = u_1,\ d_{k+1} = u_{k+1} - \sum_{i=1}^{k} \frac{u_{k+1}^T H d_i}{d_i^T H d_i},\ k = 1, \cdots, n-1.$$

试证明 $d_1, \cdots, d_n$ 关于 H 共轭.

5. 设 H 是 n 阶具有不同特征值的对称正定矩阵, 证明 H 的不同特征值对应的特征向量关于 H 是共轭的.

6. 用共轭梯度法求下列函数的极小点:

(1) $f(x) = 4x_1^2 + 4x_2^2 - 4x_1x_2 - 12x_2$, 取初始点 $x_0 = (-0.5, 1)^T$.

(2) $f(x) = x_1^2 - 2x_1x_2 + 2x_2^2 + x_3^2 - x_2x_3 + x_1 + 3x_2 - x_3$, 取初始点 $x_0 = (0, 0, 0)^T$.

7. 设 H 是 n 阶对称正定阵, 非零向量 $d_1, d_2, \cdots, d_n \in \mathbb{R}^n$ 关于矩阵 H 共轭, 证明:

(1) $x = \sum_{i=1}^{n} \frac{d_i^T H d_i}{d_i^T H d_i} d_i,\ \forall x \in \mathbb{R}^n$;　(2) $H^{-1} = \sum_{i=1}^{n} \frac{d_i d_i^T}{d_i^T H d_i}$.

8. 设 $p_1, p_n, \cdots, p_n \in \mathbb{R}^n$ 为一组线性无关向量, H 是 n 阶对称正定阵, 令向量 d_k 为:

$$d_k = \begin{cases} p_k, & k = 1 \\ p_k - \sum_{i=1}^{k-1} \frac{d_i^T H p_i}{d_i^T H d_i} d_i, & k = 2, \cdots, n \end{cases}$$

证明: $d_1, d_2, \cdots, d_n$ 关于 H 共轭.

第五章　拟牛顿法

前一章所介绍的牛顿法的优点是具有二阶收敛速度, 但当 Hesse 阵 $G(x_k) = \nabla^2 f(x_k)$ 不正定时, 不能保证所产生的方向是目标函数在 x_k 处的下降方向. 特别地, 当 $G(x_k)$ 奇异时, 算法就无法继续进行下去. 尽管修正牛顿法可以克服这一缺陷, 但其中的修正参数 μ_k 的选取很难把握, 过大或过小都会影响到收敛速度. 此外, 牛顿法的每一迭代步都需要目标函数的二阶导数, 即 Hesee 阵, 对于大规模问题其计算量是惊人的.

本章即将介绍的拟牛顿法克服了这些缺点, 并且在一定条件下这类算法仍然具有较快的收敛速度 — 超线性收敛速度.

5.1　拟牛顿法及其性质

拟牛顿法的基本思想是在基本牛顿法的步 2 中用 Hesee 阵 $G_k = \nabla^2 f(x_k)$ 的某个近似矩阵 B_k 取代 G_k. 通常, B_k 应具有下面的三个特点:

(1) 在某种意义下有 $B_k \approx G_k$, 使相应的算法产生的方向近似于牛顿方向, 以确保算法具有较快的收敛速度.

(2) 对所有的 k, B_k 是对称正定的, 从而使得算法所产生的方向是函数 f 在 x_k 处下降方向.

(3) 矩阵 B_k 更新规则相对比较简单, 即通常采用一个秩 1 或秩 2 矩阵进行校正.

下面介绍满足这三个特点的矩阵 B_k 的构造. 设 $f : \mathbb{R}^n \to \mathbb{R}$ 在开集 $D \subset \mathbb{R}^n$ 上二次连续可微. 那么, f 在 x_{k+1} 处的二次近似模型为

$$f(x) \approx f(x_{k+1}) + g_{k+1}^T(x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^T G_{k+1}(x - x_{k+1}).$$

对上式求导数得

$$g(x) \approx g_{k+1} + G_{k+1}(x - x_{k+1}).$$

令 $x = x_k$, 位移 $s_k = x_{k+1} - x_k$, 梯度差 $y_k = g_{k+1} - g_k$, 则有

$$G_{k+1}s_k \approx y_k.$$

注意到, 对于二次函数 f, 上式是精确成立的. 现在, 我们要求在拟牛顿法中构造出 Hesse 阵的近似矩阵 B_k 满足这种关系式, 即

$$B_{k+1}s_k = y_k. \tag{5.1}$$

上式通常称作拟牛顿方程或拟牛顿条件. 令 $H_{k+1} = B_{k+1}^{-1}$, 则得到拟牛顿方程的另一个形式:

$$H_{k+1}y_k = s_k, \tag{5.2}$$

其中 H_{k+1} 是 Hesse 阵逆的近似. 搜索方向由 $d_k = -H_k g_k$ 或 $B_k d_k = -g_k$ 确定. 根据 B_k(或 H_k) 的第三个特点, 可令

$$B_{k+1} = B_k + E_k, \quad H_{k+1} = H_k + D_k, \tag{5.3}$$

其中 E_k, D_k 是秩 1 或秩 2 矩阵. 通常将由拟牛顿方程 (5.1) (或 (5.2)) 和校正规则 (5.3) 所确定的方法称为拟牛顿法.

下面我们介绍一个对称秩 1 校正公式. 在 (5.3) 中取 $E_k = \alpha u_k u_k^T$ (秩 1 矩阵), 其中 $\alpha \in \mathbb{R}$, $u_k \in \mathbb{R}^n$. 由拟牛顿方程 (5.1) 得

$$(B_k + \alpha u_k u_k^T)s_k = y_k,$$

即有

$$\alpha(u_k^T s_k)u_k = y_k - B_k s_k. \tag{5.4}$$

上式表明向量 u_k 平行于 $(y_k - B_k s_k)$, 即存在常数 β 使得 $u_k = \beta(y_k - B_k s_k)$. 因此有

$$E_k = \alpha\beta^2(y_k - B_k s_k)(y_k - B_k s_k)^T.$$

于是, 由 (5.4) 得

$$\alpha\beta^2[(y_k - B_k s_k)^T s_k](y_k - B_k s_k) = (y_k - B_k s_k).$$

由此, 若 $(y_k - B_k s_k)^T s_k \neq 0$, 可取 $\alpha\beta^2[(y_k - B_k s_k)^T s_k] = 1$, 即

$$\alpha\beta^2 = \frac{1}{(y_k - B_k s_k)^T s_k}, \quad E_k = \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}.$$

故得对称秩 1 校正公式如下:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}. \tag{5.5}$$

类似地, 利用拟牛顿方程 (5.2), 对 H_k 进行对称秩 1 修正可得

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}. \tag{5.6}$$

有了对称秩 1 校正公式后, 利用它可以构造求解无约束优化问题的一个拟牛顿算法, 步骤如下:

算法 14 (**对称秩 1 算法**)

步 0　给定初始点 $x_0 \in \mathbb{R}^n$, 终止误差 $0 \le \varepsilon \ll 1$. 初始对称正定阵 H_0 (通常取单位阵 I_n). 令 $k := 0$.

步 1　若 $\|g_k\| \le \varepsilon$, 停算, 输出 x_k 作为近似极小点.

步 2　计算搜索方向 $d_k = -H_k g_k$.

步 3　用线搜索技术求步长 α_k.

步 4　令 $x_{k+1} = x_k + \alpha_k d_k$, 由对称秩 1 校正公式 (5.6) 确定 H_{k+1}.

步 5　令 $k := k + 1$, 转步 1.

下面给出基于 Armijo 搜索的对称秩 1 算法的 Matlab 程序.

程序 8 (**对称秩 1 算法程序**)

```
function [x,val,k]=sr1(fun,gfun, x0)
%功能: 用对称秩1算法求解无约束问题:  min f(x)
%输入: x0是初始点, fun, gfun分别是目标函数及其梯度
%输出:  x, val分别是近似最优点和最优值,  k是迭代次数.
maxk=500;    %给出最大迭代次数
rho=0.55;sigma=0.4; epsilon=1e-5;
k=0;    n=length(x0); Hk=eye(n);
while(k<maxk)
    gk=feval(gfun,x0); %计算梯度
    dk=-Hk*gk; %计算搜索方向
    if(norm(gk)<epsilon), break; end  %检验终止准则
    m=0; mk=0;
    while(m<20)   % 用Armijo搜索求步长
```

```
            if(feval(fun,x0+rho^m*dk)<feval(fun,x0)+sigma*rho^m*gk'*dk)
               mk=m; break;
            end
            m=m+1;
        end
        x=x0+rho^mk*dk;
        sk=x-x0;  yk=feval(gfun,x)-gk;
        Hk=Hk+(sk-Hk*yk)*(sk-Hk*yk)'/((sk-Hk*yk)'*yk); %秩1校正
        k=k+1;   x0=x;
    end
    val=feval(fun,x0);
```

例 15 利用程序 8 求解无约束优化问题

$$\min\ f(x)=100(x_1^2-x_2)^2+(x_1-1)^2,\ \ x=(x_1,x_2)^T\in\mathbb{R}^2.$$

该问题有精确解 $x^*=(1,1)^T$, $f(x^*)=0$.

解 取终止准则值为 $\|\nabla f(x_k)\|\le 10^{-5}$, 利用程序 8, 取不同的初始点, 数值结果如下表.

表 6.1 对称秩 1 校正算法的数值结果.

初始点 (x_0)	迭代次数 (k)	目标函数值 ($f(x_k)$)
$(0,0)^T$	22	7.0304×10^{-19}
$(0.5,0.5)^T$	19	3.8208×10^{-16}
$(2,2)^T$	38	3.3992×10^{-20}
$(-1,-1)^T$	45	8.2927×10^{-16}
$(1,10)^T$	98	1.9321×10^{-16}
$(10,10)^T$	142	2.1578×10^{-15}

说明 上述程序的调用方式是:

```
x0=[-1.2 1]';
[x,val,k]=sr1('fun','gfun',x0)
```

其中 fun, gfun 分别是求目标函数值及其梯度的 M 函数文件.

5.2　BFGS 算法及其 Matlab 实现

BFGS 校正是目前最流行也是最有效的拟牛顿校正, 它是由 Broyden, Fletcher, Goldfarb 和 Shanno 在 1970 年各自独立提出的拟牛顿法, 故称为 BFGS 算法. 其基本思想是: 在 (5.3) 中取修正矩阵 E_k 为秩 2 矩阵:

$$E_k = \alpha u_k u_k^T + \beta v_k v_k^T,$$

其中 $u_k, v_k \in \mathbb{R}^n$ 是待定向量, $\alpha, \beta \in \mathbb{R}$ 是待定实数. 于是由拟牛顿方程 (5.1) 可得

$$(B_k + \alpha u_k u_k^T + \beta v_k v_k^T)s_k = y_k,$$

或等价地

$$\alpha(u_k^T s_k)u_k + \beta(v_k^T s_k)v_k = y_k - B_k s_k. \tag{5.7}$$

不难发现, 满足上式的向量 u_k 和 v_k 不唯一, 可取 u_k 和 v_k 分别平行于 $B_k s_k$ 和 y_k, 即令 $u_k = \gamma B_k s_k$, $v_k = \theta y_k$, 其中 γ, θ 是待定的参数. 于是我们有

$$E_k = \alpha\gamma^2 B_k s_k s_k^T B_k + \beta\theta^2 y_k y_k^T.$$

将 u_k, v_k 的表达式代入 (5.7) 得

$$\alpha[(\gamma B_k s_k)^T s_k](\gamma B_k s_k) + \beta[(\theta y_k)^T s_k)(\theta y_k) = y_k - B_k s_k,$$

整理得

$$[\alpha\gamma^2(s_k^T B_k s_k) + 1]B_k s_k + [\beta\theta^2(y_k^T s) - 1]y_k = 0.$$

故此, 可令 $\alpha\gamma^2(s_k^T B_k s_k) + 1 = 0$ 及 $\beta\theta^2(y_k^T s) - 1 = 0$, 即

$$\alpha\gamma^2 = -\frac{1}{s_k^T B_k s_k}, \quad \beta\theta^2 = \frac{1}{y_k^T s_k}.$$

从而得到如下的 BFGS 秩 2 修正公式如下

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \tag{5.8}$$

显然, 由 (5.8) 可知, 若 B_k 对称, 校正后的 B_{k+1} 也对称, 并且可以证明 BFGS 校正公式的如下性质:

引理 4 设 B_k 对称正定, B_{k+1} 由 BFGS 校正公式 (5.8) 确定, 那么 B_{k+1} 对称正定的充要条件是 $y_k^T s_k > 0$.

证 必要性是显然的. 因 $y_k^T s_k = s_k^T B_{k+1} s_k$, 故若 B_{k+1} 正定, 则显然有 $y_k^T s_k > 0$.

下面证明充分性. 设 $y_k^T s_k > 0$ 且 B_k 正定. 由校正公式 (5.8), 对任意的 $0 \neq d \in \mathbb{R}^n$, 我们有

$$d^T B_{k+1} d = d^T B_k d - \frac{(d^T B_k s)^2}{s_k^T B_k s_k} + \frac{(d^T y_k)^2}{y_k^T s_k}. \tag{5.9}$$

因 B_k 对称正定, 故存在对称正定阵 $B_k^{1/2}$, 使得 $B_k = B_k^{1/2} B_k^{1/2}$. 从而, 利用 Cauchy-Schwarz 不等式得

$$\begin{aligned}(d^T B_k s_k)^2 &= [(B_k^{1/2} d)^T (B_k^{1/2} s_k)]^2 \leq \|B_k^{1/2} d\|^2 \|B_k^{1/2} s_k\|^2 \\ &= (B_k^{1/2} d)^T (B_k^{1/2} d) \cdot (B_k^{1/2} s_k)^T (B_k^{1/2} s_k) \\ &= (d^T B_k d)(s_k^T B_k s_k). \end{aligned} \tag{5.10}$$

不难发现, 上式成立等式的充要条件是存在实数 $\tau_k \neq 0$, 使得 $B_k^{1/2} d = \tau_k B_k^{1/2} s_k$, 即 $d = \tau_k s_k$.

故而, 若不等式 (5.10) 严格成立, 则由 (5.9) 得

$$d^T B_{k+1} d > d^T B_k d - \frac{(d^T B_k d)(s_k^T B_k s_k)}{s_k^T B_k s_k} + \frac{(d^T y_k)^2}{y_k^T s_k} > 0.$$

否则, 若 (5.10) 等式成立, 即存在 τ_k, 使得 $d = \tau_k s_k$, 则由 (5.9), (5.10) 得

$$d^T B_{k+1} d = \frac{(d^T y_k)^2}{y_k^T s_k} = \frac{\tau_k^2 (s_k^T y_k)^2}{y_k^T s_k} = \tau_k^2 y_k^T s_k > 0.$$

故对任意的 $d \in \mathbb{R}^n$, $d \neq 0$, 总有 $d^T B_{k+1} d > 0$. 证毕. □

上面的引理表明, 若初始矩阵 B_0 对称正定且在迭代过程中保持 $y_k^T s_k > 0$, $\forall\, k \geq 0$, 则由 BFGS 校正公式产生的矩阵序列 $\{B_k\}$ 是对称正定的. 从而方程组 $B_k d = -g_k$ 有唯一解 d_k, 且 d_k 是函数 f 在 x_k 处的下降方向.

引理 5 若在 BFGS 算法中采用精确线搜索或 Wolfe 搜索准则, 则有 $y_k^T s_k > 0$.

证 注意到对于精确线搜索有 $g_{k+1}^T d_k = 0$. 故

$$y_k^T s_k = \alpha_k (g_{k+1} - g_k)^T d_k = -\alpha_k g_k^T d_k > 0.$$

对于 Wolfe 搜索准则, 利用该准则的第二个不等式 (即 $\nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k$), 得

$$\begin{aligned} y_k^T s_k &= \alpha_k (g_{k+1} - g_k)^T d_k \geq \alpha_k (\sigma - 1) g_k^T d_k \\ &= -\alpha_k (1 - \sigma) g_k^T d_k > 0. \end{aligned}$$

证毕. □

已有证明表示, Armijo 搜索准则一般不能保证 $y_k^T s_k > 0$. 但 Armijo 准则因其简单且易于程序实现而深得人们的喜爱, 因此, 为了保证采用 Armijo 准则时矩阵序列 $\{B_k\}$ 的对称正定性, 可采用如下的校正方式

$$B_{k+1} = \begin{cases} B_k, & \text{若 } y_k^T s_k \leq 0, \\ B_k - \dfrac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \dfrac{y_k y_k^T}{y_k^T s_k}, & \text{若 } y_k^T s_k > 0. \end{cases} \tag{5.11}$$

不难发现, 只要 B_0 对称正定, 上述校正公式可以保证矩阵序列 $\{B_k\}$ 的对称正定性. 下面给出基于 Armijo 搜索准则的 BFGS 算法的详细步骤.

算法 15 (**BFGS 算法**)

步 0 给定参数 $\delta \in (0,1)$, $\sigma \in (0, 0.5)$, 初始点 $x_0 \in \mathbb{R}^n$, 终止误差 $0 \leq \varepsilon \ll 1$. 初始对称正定阵 B_0 (通常取为 $G(x_0)$ 或单位阵 I_n). 令 $k := 0$.

步 1 计算 $g_k = \nabla f(x_k)$. 若 $\|g_k\| \leq \varepsilon$, 停算, 输出 x_k 作为近似极小点.

步 2 解线性方程组得解 d_k:

$$B_k d = -g_k. \tag{5.12}$$

步 3 设 m_k 是满足下列不等式的最小非负整数 m:

$$f(x_k + \delta^m d_k) \leq f(x_k) + \sigma \delta^m g_k^T d_k. \tag{5.13}$$

令 $\alpha_k = \delta^{m_k}$, $x_{k+1} = x_k + \alpha_k d_k$.

步 4 由校正公式 (5.11) 确定 B_{k+1}.

步 5 令 $k := k + 1$, 转步 1.

下面给出基于 Armijo 搜索的 BFGS 算法的 Matlab 程序.

程序 9（BFGS 算法程序）

```
function [x,val,k]=bfgs(fun,gfun,x0,varargin)
%功能: 用BFGS算法求解无约束问题:  min f(x)
%输入: x0是初始点, fun, gfun分别是目标函数及其梯度;
% varargin是输入的可变参数变量, 简单调用bfgs时可以忽略它,
% 但若其它程序循环调用该程序时将发挥重要的作用
%输出:  x, val分别是近似最优点和最优值, k是迭代次数.
maxk=500;   %给出最大迭代次数
rho=0.55;sigma=0.4; epsilon=1e-5;
k=0;   n=length(x0);
Bk=eye(n);  %Bk=feval('Hess',x0);
while(k<maxk)
    gk=feval(gfun,x0,varargin{:}); %计算梯度
    if(norm(gk)<epsilon), break; end  %检验终止准则
    dk=-Bk\gk;  %解方程组, 计算搜索方向
    m=0; mk=0;
    while(m<20)   % 用Armijo搜索求步长
        newf=feval(fun,x0+rho^m*dk,varargin{:});
        oldf=feval(fun,x0,varargin{:});
        if(newf<oldf+sigma*rho^m*gk'*dk)
            mk=m; break;
        end
        m=m+1;
    end
    %BFGS校正
    x=x0+rho^mk*dk;
    sk=x-x0;  yk=feval(gfun,x,varargin{:})-gk;
    if(yk'*sk>0)
        Bk=Bk-(Bk*sk*sk'*Bk)/(sk'*Bk*sk)+(yk*yk')/(yk'*sk);
    end
    k=k+1;   x0=x;
end
val=feval(fun,x0,varargin{:});
```

例 16 利用程序 9 求解无约束优化问题

$$\min\ f(x)=100(x_1^2-x_2)^2+(x_1-1)^2,\ \ x=(x_1,x_2)^T\in\mathbb{R}^2.$$

该问题有精确解 $x^* = (1,1)^T$, $f(x^*) = 0$.

解 取终止准则值为 $\|\nabla f(x_k)\| \le 10^{-5}$, 利用程序 9, 取不同的初始点, 数值结果如下表.

表 6.2 BFGS 校正算法的数值结果.

初始点 (x_0)	迭代次数 (k)	目标函数值 ($f(x_k)$)
$(0,0)^T$	20	2.2005×10^{-11}
$(0.5,0.5)^T$	15	1.946×10^{-16}
$(2,2)^T$	24	2.1171×10^{-15}
$(-1,-1)^T$	31	1.3594×10^{-12}
$(1,10)^T$	36	1.3757×10^{-15}
$(10,10)^T$	66	6.3531×10^{-15}
$(-1.2,1)^T$	32	6.7539×10^{-16}

从上表可以看出, BFGS 算法比对称秩 1 算法更为有效.

5.3 DFP 算法及其 Matlab 实现

DFP 校正是第一个拟牛顿校正, 是 1959 年由 Davidon 提出的, 后经 Fletcher 和 Powell 解释和改进, 故名之为 DFP 算法, 它也是求解无约束优化问题最有效的算法之一. 类似于 BFGS 校正公式的推导, 可得 DFP 校正公式如下:

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{s_k^T y_k}. \tag{5.14}$$

同样, 不难发现, 由 (5.14), 若 H_k 对称, 校正后的 H_{k+1} 也对称, 并且类似于引理 4 的证明, 可得 DFP 校正公式的如下性质:

引理 6 设 H_k 对称正定, H_{k+1} 由 DFP 校正公式 (5.14) 确定, 那么 H_{k+1} 对称正定的充要条件是 $s_k^T y_k > 0$.

类似于引理 5, 可以证明, 当采用精确线搜索或 Wolfe 搜索准则时, 矩阵序列 $\{H_k\}$ 的正定性条件 $s_k^T y_k > 0$ 可以被满足. 但一般来说, Armijo 搜索准备不能满

足这一条件, 需要作如下修正:

$$H_{k+1}=\begin{cases}H_k, & 若\ s_k^Ty_k\le 0,\\ H_k-\dfrac{H_ky_ky_k^TH_k}{y_k^TH_ky_k}+\dfrac{s_ks_k^T}{s_k^Ty_k}, & 若\ y_k^Ts_k>0.\end{cases}\tag{5.15}$$

下面给出基于 Armijo 搜索准则的 DFP 算法的详细步骤.

算法 16 (**DFP 算法**)

步 0 给定参数 $\delta\in(0,1)$, $\sigma\in(0,0.5)$, 初始点 $x_0\in\mathbb{R}^n$, 终止误差 $0\le\varepsilon\ll 1$. 初始对称正定阵 H_0 (通常取为 $G(x_0)^{-1}$ 或单位阵 I_n). 令 $k:=0$.

步 1 计算 $g_k=\nabla f(x_k)$. 若 $\|g_k\|\le\varepsilon$, 停算, 输出 x_k 作为近似极小点.

步 2 计算搜索方向:

$$d_k=-H_kg_k.\tag{5.16}$$

步 3 设 m_k 是满足下列不等式的最小非负整数 m:

$$f(x_k+\delta^md_k)\le f(x_k)+\sigma\delta^mg_k^Td_k.\tag{5.17}$$

令 $\alpha_k=\delta^{m_k}$, $x_{k+1}=x_k+\alpha_kd_k$.

步 4 由校正公式 (5.15) 确定 H_{k+1}.

步 5 令 $k:=k+1$, 转步 1.

下面给出基于 Armijo 搜索的 DFP 算法的 Matlab 程序.

程序 10 (**DFP 算法程序**)

```
function [x,val,k]=dfp(fun,gfun,x0)
%功能: 用DFP算法求解无约束问题: min f(x)
%输入: x0是初始点, fun, gfun分别是目标函数及其梯度
%输出: x, val分别是近似最优点和最优值, k是迭代次数.
maxk=1e5;  %给出最大迭代次数
rho=0.55;sigma=0.4; epsilon=1e-5;
k=0;   n=length(x0);
Hk=inv(feval('Hess',x0));   %Hk=eye(n);
while(k<maxk)
    gk=feval(gfun,x0); %计算梯度
    if(norm(gk)<epsilon), break; end  %检验终止准则
```

```
        dk=-Hk*gk;  %解方程组, 计算搜索方向
        m=0; mk=0;
        while(m<20)   % 用Armijo搜索求步长
            if(feval(fun,x0+rho^m*dk)<feval(fun,x0)+sigma*rho^m*gk'*dk)
                mk=m; break;
            end
            m=m+1;
        end
        %DFP校正
        x=x0+rho^mk*dk;
        sk=x-x0;  yk=feval(gfun,x)-gk;
        if(sk'*yk>0)
            Hk=Hk-(Hk*yk*yk'*Hk)/(yk'*Hk*yk)+(sk*sk')/(sk'*yk);
        end
        k=k+1;     x0=x;
    end
    val=feval(fun,x0);
```

例 17 利用程序 10 求解无约束优化问题

$$\min\ f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2,\ \ x = (x_1, x_2)^T \in \mathbb{R}^2.$$

该问题有精确解 $x^* = (1,1)^T$, $f(x^*) = 0$.

解 取终止准则值为 $\|\nabla f(x_k)\| \le 10^{-5}$, 利用程序 10, 取不同的初始点, 数值结果如下表.

表 6.3 DFP 校正算法的数值结果.

初始点 (x_0)	迭代次数 (k)	目标函数值 ($f(x_k)$)
$(0,0)^T$	23	9.4910×10^{-16}
$(0.5,0.5)^T$	19	1.5488×10^{-15}
$(2,2)^T$	22	4.0247×10^{-13}
$(-1,-1)^T$	35	2.2338×10^{-12}
$(1,10)^T$	1	0
$(10,10)^T$	77	8.6152×10^{-20}
$(-1.2,1)^T$	34	3.0415×10^{-14}

从上表可以看出, DFP 算法的计算效率似乎不如 BFGS 算法.

5.4 Broyden 族算法及其 Matlab 实现

前面我们讨论了 BFGS 和 DFP 校正, 它们都是由 y_k 和 B_ks_k (或 s_k 和 H_ky_k) 组成的秩 2 校正. 本节讨论由 BFGS 和 DFP 校正的凸组合产生的一类校正族

$$B_{k+1}^{\theta} = \theta_k B_{k+1}^{\rm DFP} + (1-\theta_k)B_{k+1}^{\rm BFGS} \tag{5.18}$$

$$= B_k - \frac{B_ks_ks_k^TB_k}{s_k^TB_ks_k} + \frac{y_ky_k^T}{s_k^Ty_k} + \theta_k u_k u_k^T, \tag{5.19}$$

其中 θ_k 为实参数, u_k 由下式定义:

$$u_k = \sqrt{s_k^TB_ks_k}\left(\frac{y_k}{y_k^Ts_k} - \frac{B_ks_k}{s^TB_ks_k}\right). \tag{5.20}$$

这类校正公式称为 Broyden 族. 不难发现, 在 (5.19) 中, 当 $\theta_k = 0$, 即得到 BFGS 公式; 当 $\theta_k = 1$, 即得到 DFP 公式.

对应地, 关于 H_k 的 Broyden 族校正公式为

$$H_{k+1}^{\phi} = \phi_k H_{k+1}^{\rm BFGS} + (1-\phi_k)H_{k+1}^{\rm DFP} \tag{5.21}$$

$$= H_k - \frac{H_ky_ky_k^TH_k}{y_k^TH_ky_k} + \frac{s_ks_k^T}{s_k^Ty_k} + \phi_k v_k v_k^T, \tag{5.22}$$

其中 ϕ_k 为实参数, v_k 由下式定义:

$$v_k = \sqrt{y_k^TH_ky_k}\left(\frac{s_k}{y_k^Ts_k} - \frac{H_ky_k}{y_k^TH_ky_k}\right). \tag{5.23}$$

可以证明参数 θ_k 与 ϕ_k 之间的关系为

$$\theta_k = \frac{1-\phi_k}{1-\phi_k(1-\mu_k)}, \tag{5.24}$$

其中

$$\mu_k = \frac{(s_k^TB_ks_k)(y_k^TH_ky_k)}{(s_k^Ty_k)^2}.$$

不难发现 $u_k^Ts_k = 0$ 和 $v_k^Ty_k = 0$, 因此 Broyden 族 (5.19) 和 (5.22) 给出的校正公式于任何参数 θ_k 和 ϕ_k 都满足拟牛顿方程 (5.1) 和 (5.2).

下面我们给出 Broyden 族校正公式的两个性质.

定理 25 用 Broyden 族算法求解极小化二次目标函数问题

$$\min f(x) = \frac{1}{2}x^T Gx + b^T x + c. \tag{5.25}$$

若果初始矩阵 H_0 是正定的, 算法所产生的迭代点列是互异的. 则

(1) 当 $y_k^T s_k > 0$ 且 $\phi_k \geq 0$ 或 $\theta_k \geq 0$ 时, Broyden 族校正公式保持正定性.

(2) 算法所产生的搜索方向 $d_0, d_1, \cdots, d_k (k \leq n-1)$ 满足

(a) $d_i^T G d_j = 0,\ 0 \leq i < j \leq k;$ (b) $H_k^\phi y_i = s_i,\ 0 \leq i \leq k-1.$

证 对 k 用归纳法. 注意到 $s_i = x_{i+1} - x_i = \alpha_i d_i$, $y_i = g_{i+1} - g_i = G(x_{i+1} - x_i) = Gs_i$. 那么, 当 $k = 1$ 时, 由拟牛顿方程 (5.2) 可知

$$H_1^\phi y_0 = s_0$$

成立, 且有

$$\begin{aligned} d_0^T G d_1 &= (Gd_0)^T d_1 = -\frac{1}{\alpha_0}(Gs_0)^T H_1^\phi g_1 \\ &= -\frac{1}{\alpha_0} y_0^T H_1^\phi g_1 = -\frac{1}{\alpha_0} s_0^T g_1 \\ &= -d_0^T g_1 = 0, \end{aligned}$$

即 $k = 1$ 时结论成立.

设 $k = l$ 时结论成立, 要证 $k = l + 1$ 时结论也成立. 由归纳法假设有

$$d_i^T G d_j = 0,\ \ 0 \leq i < j \leq l; \quad H_l^\phi y_i = s_i,\ \ 0 \leq i \leq l-1. \tag{5.26}$$

当 $k = l + 1$ 时, 对于 $0 \leq i \leq l - 1$, 我们有

$$\begin{aligned} H_{l+1}^\phi y_i &= \left(H_l^\phi - \frac{H_l^\phi y_l y_l^T H_l^\phi}{y_l^T H_l^\phi y_l} + \frac{s_l s_l^T}{s_l^T y_l} + \phi v_l v_l^T\right) y_i \\ &= H_l^\phi y_i - \frac{H_l^\phi y_l y_l^T H_l^\phi y_i}{y_l^T H_l^\phi y_l} + \frac{s_l s_l^T y_i}{s_l^T y_l} + \phi v_l v_l^T y_i \\ &= s_i - \frac{H_l^\phi y_l (y_l^T s_i)}{y_l^T H_l^\phi y_l} + \frac{s_l s_l^T y_i}{s_l^T y_l} + \phi v_l v_l^T y_i \\ &= s_i - \frac{H_l^\phi y_l (s_l^T G s_i)}{y_l^T H_l^\phi y_l} + \frac{s_l s_l^T G s_i}{s_l^T y_l} + \phi v_l v_l^T y_i \end{aligned}$$

由 $s_i = \alpha_i d_i$ 和 (5.26) 的第一式可知, 上式等式右边的第二、三两项为 0. 下面考虑第三项. 注意到

$$
\begin{aligned}
\frac{\phi}{(y_l^T H_l^\phi y_l)} v_l v_l^T y_i &= \left(\frac{s_l}{y_l^T s_l} - \frac{H_l^\phi y_l}{y_l^T H_l^\phi y_l}\right)\left(\frac{s_l}{y_l^T s_l} - \frac{H_l^\phi y_l}{y_l^T H_l^\phi y_l}\right)^T y_i \\
&= \frac{s_l s_l^T y_i}{(y_l^T s_l)^2} + \frac{H_l^\phi y_l y_l^T H_l^\phi y_i}{(y_l^T H_l^\phi y_l)^2} - \frac{s_l y_l^T H_l^\phi y_i}{(y_l^T s_l)(y_l^T H_l^\phi y_l)} - \frac{H_l^\phi y_l s_l^T y_i}{(y_l^T s_l)(y_l^T H_l^\phi y_l)} \\
&= \frac{s_l (s_l^T G s_i)}{(y_l^T s_l)^2} + \frac{H_l^\phi y_l y_l^T s_i}{(y_l^T H_l^\phi y_l)^2} - \frac{s_l y_l^T s_i}{(y_l^T s_l)(y_l^T H_l^\phi y_l)} - \frac{H_l^\phi y_l (s_l^T G s_i)}{(y_l^T s_l)(y_l^T H_l^\phi y_l)} \\
&= 0 + \frac{H_l^\phi y_l (s_l^T G s_i)}{(y_l^T H_l^\phi y_l)^2} - \frac{s_l (s_l^T G s_i)}{(y_l^T s_l)(y_l^T H_l^\phi y_l)} - 0 = 0.
\end{aligned}
$$

故有

$$
H_{l+1}^\phi y_i = s_i, \quad 0 \le i \le l-1. \tag{5.27}
$$

又因为拟牛顿方程 $H_{l+1}^\phi y_l = s_l$ 满足, 所以 (5.27) 对于 $0 \le i \le l$ 成立.

下面证明

$$
d_i^T G d_j = 0, \quad 0 \le i < j \le l+1. \tag{5.28}
$$

由归纳法假设 (5.26), 只需证明对于 $0 \le i \le l$ 成立

$$
d_i G d_{l+1} = 0.
$$

事实上,

$$
\begin{aligned}
d_i G d_{l+1} &= (G d_i)^T d_{l+1} = \frac{1}{\alpha_i} y_i^T \big(-H_{l+1}^\phi g_{l+1}\big) \\
&= -\frac{1}{\alpha_i} s_i^T g_{l+1} = -d_i^T g_{l+1} \\
&= -d_i^T \Big(g_{i+1} + \sum_{j=i+1}^{l} (g_{j+1} - g_j)\Big) \\
&= -g_{i+1}^T d_i + \sum_{j=i+1}^{l} y_j^T d_i \\
&= -g_{i+1}^T d_i + \sum_{j=i+1}^{l} s_j^T G d_i = 0.
\end{aligned}
$$

因此, 对于 $k = l+1$, 结论成立. □

推论 2 在定理 25 的条件下, 有如下结论:

(1) Broyden 族校正公式至多迭代 n 次就可以达到极小点 x^*, 即存在 $k\,(0 \le k \le n)$, 使得 $x_k = x^*$.

(2) 若 $x_k \ne x^*$, $0 \le k \le n-1$, 则 $H_n = G^{-1}$.

证 由定理 25 可知, Broyden 族校正算法是一种共轭方向法, 故结论 (1) 成立.

若 $x_k \ne x^*$, $0 \le k \le n-1$, 则 Broyden 族校正公式产生 n 个共轭方向 $d_0, d_1, \cdots, d_{n-1}$, 因而它们是线性无关的, 从而 $s_0, s_1, \cdots, s_{n-1}$ 也是线性无关的. 又由定理 25 得

$$H_n G s_i = H_n y_i = s_i, \ \ i = 0, 1, \cdots, n-1,$$

即

$$H_n G[s_0, s_1, \cdots, s_{n-1}] = [s_0, s_1, \cdots, s_{n-1}].$$

因矩阵 $[s_0, s_1, \cdots, s_{n-1}]$ 非奇异, 故有 $H_n G = I$, 即 $H_n = G^{-1}$. 证毕. □

下面给出基于 Armijo 搜索准则的 Broyden 族算法的详细步骤.

算法 17 (Broyden 族算法)

步 0 给定参数 $\delta \in (0,1)$, $\sigma \in (0, 0.5)$,$\phi \in [0,1]$. 初始点 $x_0 \in \mathbb{R}^n$, 终止误差 $0 \le \varepsilon \ll 1$. 初始对称正定阵 H_0 (通常取为 $G(x_0)^{-1}$ 或单位阵 I_n). 令 $k := 0$.

步 1 计算 $g_k = \nabla f(x_k)$. 若 $\|g_k\| \le \varepsilon$, 停算, 输出 x_k 作为近似极小点.

步 2 计算搜索方向:

$$d_k = -H_k g_k. \tag{5.29}$$

步 3 设 m_k 是满足下列不等式的最小非负整数 m:

$$f(x_k + \delta^m d_k) \le f(x_k) + \sigma \delta^m g_k^T d_k. \tag{5.30}$$

令 $\alpha_k = \delta^{m_k}$, $x_{k+1} = x_k + \alpha_k d_k$.

步 4 由下面的校正公式确定 H_{k+1} :

$$H_{k+1} = \begin{cases} H_k, & \text{若 } s_k^T y_k \le 0, \\ H_k - \dfrac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \dfrac{s_k s_k^T}{s_k^T y_k} + \phi v_k v_k^T, & \text{若 } y_k^T s_k > 0. \end{cases} \tag{5.31}$$

其中, v_k 由下式定义:

$$v_k = \sqrt{y_k^T H_k y_k}\left(\frac{s_k}{y_k^T s_k} - \frac{H_k y_k}{y_k^T H_k y_k}\right).$$

步 5 令 $k := k+1$, 转步 1.

下面给出基于 Armijo 搜索的 Broyden 族算法的 Matlab 程序.

程序 11 (Broyden 族算法程序)

```
function [x,val,k]=broyden(fun,gfun,x0)
%功能: 用Broyden族算法求解无约束问题:  min f(x)
%输入: x0是初始点, fun, gfun分别是目标函数及其梯度
%输出: x,val分别是近似最优点和最优值,  k是迭代次数.
maxk=1e5;   %给出最大迭代次数
rho=0.55;sigma=0.4; epsilon=1e-5;
phi=0.5; k=0;   n=length(x0);
Hk=inv(feval('Hess',x0));   %Hk=eye(n);
while(k<maxk)
    gk=feval(gfun,x0); %计算梯度
    if(norm(gk)<epsilon), break; end  %检验终止准则
    dk=-Hk*gk;  %解方程组, 计算搜索方向
    m=0; mk=0;
    while(m<20)   % 用Armijo搜索求步长
        if(feval(fun,x0+rho^m*dk)<feval(fun,x0)+sigma*rho^m*gk'*dk)
            mk=m; break;
        end
        m=m+1;
    end
    %Broyden族校正
    x=x0+rho^mk*dk;
    sk=x-x0;  yk=feval(gfun,x)-gk;
    Hy=Hk*yk; sy=sk'*yk;  yHy=yk'*Hk*yk;
    if(sy<0.2*yHy)
        theta=0.8*yHy/(yHy-sy);
        sk=theta*sk+(1-theta)*Hy;
        sy=0.2*yHy;
    end
```

```
    vk=sqrt(yHy)*(sk/sy - Hy/yHy);
    Hk=Hk-(Hy*Hy')/yHy+(sk*sk')/sy+phi*vk*vk';
    k=k+1;  x0=x;
end
val=feval(fun,x0);
```

例 18 利用程序 11 求解无约束优化问题

$$\min\ f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2,\ \ x = (x_1, x_2)^T \in \mathbb{R}^2.$$

该问题有精确解 $x^* = (1,1)^T$, $f(x^*) = 0$.

解 取终止准则值为 $\|\nabla f(x_k)\| \le 10^{-5}$, 利用程序 11, 取不同的初始点, 数值结果如下表.

表 6.4 Broyden 族校正算法的数值结果.

初始点 (x_0)	迭代次数 (k)	目标函数值 ($f(x_k)$)
$(0,0)^T$	23	9.4910×10^{-16}
$(0.5,0.5)^T$	19	1.5488×10^{-15}
$(2,2)^T$	22	4.0247×10^{-13}
$(-1,-1)^T$	35	2.2338×10^{-12}
$(1,10)^T$	1	0
$(10,10)^T$	77	8.6152×10^{-20}
$(-1.2,1)^T$	34	3.0415×10^{-14}

5.5 拟牛顿法的收敛性

本节讨论拟牛顿法的收敛性, 主要给出基于非精确线搜索的 BFGS 算法的全局收敛性和局部超线性收敛性定理. 为了方便, 我们将 BFGS 算法的迭代公式复述如下:

$$x_{k+1} = x_k - \alpha_k B_k^{-1} g_k, \tag{5.32}$$

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \tag{5.33}$$

其中, $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$, α_k 由非精确线搜索方法得到. 在讨论收敛性之前, 我们先给出如下假设条件.

假设 1 (1) 函数 $f(x)$ 在 $\mathbb{R}^n$ 上二阶连续可微.

(2) 水平集

$$\mathcal{L}(x_0)=\{x\in\mathbb{R}^n \mid f(x)\le f(x_0)\}$$

是凸集, 且函数 f 在 $\mathcal{L}(x_0)$ 上一致凸, 即存在常数 $0<m\le M$, 使得

$$m\|d\|^2\le d^TG(x)d\le M\|d\|^2. \tag{5.34}$$

(3) 存在 x^* 的一个邻域 $N(x^*,\delta)$, 使得 $G(x)=\nabla^2 f(x)$ 在该邻域内 Lipschitz 连续, 即存在常数 $L>0$, 使得

$$\|G(x)-G(x^*)\|\le L\|x-x^*\|,\quad \forall\, x\in N(x^*,\delta).$$

在后面的分析中我们需要用到一个求秩 2 校正矩阵行列式的公式, 先把它来写在下面以方便使用:

$$\det\big(I+w_1w_2^T+w_3w_4^T\big)=(1+w_1^Tw_2)(1+w_3^Tw_4)-(w_1^Tw_4)(w_2^Tw_3), \tag{5.35}$$

这里 $w_i\,(i=1,\cdots,4)$ 是任意的 n 维向量.

我们有下面的全局收敛性定理.

定理 26 设 $\{B_k\}$ 是由 BFGS 校正公式 (5.33) 产生的非奇异矩阵序列, α_k 为满足 Armijo 准则 (2.13) 的步长. 若 $f(x)$ 满足假设 1 (1) 和 (2). 那么由迭代公式 (5.32) 产生的序列 $\{x_k\}$ 全局收敛到 $f(x)$ 的极小点 x^*.

证 根据 Armijo 准则下的线搜索法的收敛定理 11, 我们只需验证搜索方向 d_k 与负梯度方向 $-g_k$ 的夹角 θ_k 满足条件 (2.15), 即 $0<\theta_k\le\dfrac{\pi}{2}-\mu$, $\mu\in\Big(0,\dfrac{\pi}{2}\Big)$. 注意到

$$\cos\theta_k=\frac{-d_k^Tg_k}{\|d_k\|\|g_k\|}=\frac{s_k^T(B_ks_k)}{\|s_k\|\|B_ks_k\|}.$$

以下只需证明由上式定义的 θ_k 满足 $\cos\theta_k\ge\delta>0$ 即可.

由 y_k 的定义，可得

$$y_k=g_{k+1}-g_k=\int_0^1 G(x_k+\tau s_k)s_k\mathrm{d}\tau, \tag{5.36}$$

故有

$$y_k^Ts_k=\int_0^1 s_k^TG(x_k+\tau s_k)s_k\mathrm{d}\tau.$$

利用假设 1 (2) 可得

$$y_k^T s_k \geq m\|s_k\|^2, \quad 即 \quad a_k \triangleq \frac{y_k^T s_k}{\|s_k\|^2} \geq m. \tag{5.37}$$

由 (5.36) 得

$$\|y_k\| \leq \int_0^1 \|G(x_k+\tau s_k)s_k\|\mathrm{d}\tau \leq \|s_k\| \int_0^1 \|G(x_k+\tau s_k)\|\mathrm{d}\tau. \tag{5.38}$$

不难发现, 由假设 1 (2) 有

$$\max_{x\in\mathcal{L}(x_0)} \|G(x)\| = \max_{x\in\mathcal{L}(x_0)} \sup_{d\neq 0} \frac{|d^T G(x) d|}{\|d\|^2} \leq M.$$

因此由 (5.38) 可推得 $\|y_k\| \leq M\|s_k\|$. 结合 (5.37), 我们有

$$b_k \triangleq \frac{\|y_k\|^2}{y_k^T s_k} \leq \frac{\|y_k\|^2}{m\|s_k\|^2} \leq \frac{M^2}{m} \triangleq \bar{b}.$$

对 BFGS 公式

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

两边求迹得

$$\mathrm{tr}(B_{k+1}) = \mathrm{tr}(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \frac{\|y_k\|^2}{y_k^T s_k}. \tag{5.39}$$

为了便于应用公式 (5.35), 我们将 BFGS 校正公式写成如下形式:

$$\begin{aligned} B_{k+1} &= B_k\Big(I + \Big[-\frac{1}{s_k^T B_k s_k} s_k\Big](s_k^T B_k) + \Big[\frac{1}{s_k^T y_k} B_k^{-1} y_k\Big] y_k^T\Big) \\ &\triangleq B_k\big(I + w_1 w_2^T + w_3 w_4^T\big). \end{aligned}$$

利用公式 (5.35) 得

$$\det\big(B_{k+1}\big) = -\det\big(B_k\big)\big(w_1^T w_4\big) = \det\big(B_k\big)\frac{s_k^T y_k}{s_k^T B_k s_k}. \tag{5.40}$$

记

$$c_k \triangleq \frac{s_k^T B_k s_k}{\|s_k\|^2},$$

则

$$\frac{\|B_k s_k\|^2}{s_k^T B_k s_k} = \left(\frac{\|s_k\|\|B_k s_k\|}{s_k^T B_k s_k}\right)^2 \frac{s_k^T B_k s_k}{\|s_k\|^2} = \frac{c_k}{\cos^2\theta_k}. \tag{5.41}$$

于是由 (5.40) 有

$$\det(B_{k+1}) = \det(B_k)\frac{s_k^T y_k}{\|s_k\|^2}\frac{\|s_k\|^2}{s_k^T B_k s_k} = \det(B_k)\frac{a_k}{c_k}. \tag{5.42}$$

关于对称正定矩阵 B, 定义函数

$$\phi(B) = \mathrm{tr}(B) - \ln(\det(B)),$$

则有 $\phi(B) > 0$. 事实上, 设 $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ 为 B 的特征值, 则

$$\begin{aligned}\phi(B) &= \mathrm{tr}(B) - \ln(\det(B)) \\ &= \sum_{i=1}^{n}\lambda_i - \ln(\lambda_1\lambda_2\cdots\lambda_n) \\ &= \sum_{i=1}^{n}(\lambda_i - \ln\lambda_i) > 0.\end{aligned}$$

由 (5.39), (5.41) 和 (5.42) 得

$$\begin{aligned}\phi(B_{k+1}) &= \mathrm{tr}(B_{k+1}) - \ln(\det(B_{k+1})) \\ &= \mathrm{tr}(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \frac{\|y_k\|^2}{y_k^T s_k} - \ln\left(\det(B_k)\frac{a_k}{c_k}\right) \\ &= \phi(B_k) - \frac{c_k}{\cos^2\theta_k} + b_k - \ln a_k + \ln c_k \\ &= \phi(B_k) + (b_k - \ln a_k - 1) \\ &\quad + \left(1 - \frac{c_k}{\cos^2\theta_k} + \ln\frac{c_k}{\cos^2\theta_k}\right) + \ln\cos^2\theta_k \\ &\le \phi(B_k) + \sigma + \ln\cos^2\theta_k,\end{aligned}$$

上式的最后一个不等式利用了函数 $\psi(t) = 1 - t + \ln t$ 在区间 $(0,\infty)$ 上的非正性及 $a_k \ge m$ 和 $b_k \le \bar{b}$, 且正常数 $\sigma \ge \bar{b} - \ln m - 1 \ge b_k - \ln a_k - 1$. 于是有

$$0 < \phi(B_{k+1}) \le \phi(B_0) + \sigma(k+1) + \sum_{i=0}^{k}\ln\cos^2\theta_i. \tag{5.43}$$

下面证明数列 $\{\cos\theta_k\} \not\to 0\,(k\to\infty)$. 用反证法. 若结论不成立, 则对上述的常数 $l > 0$, 存在 $k_0 > 0$ 使得对所有的 $k \ge k_0$, 有

$$\ln\cos^2\theta_k < -2\sigma.$$

由 (5.43) 得

$$\begin{aligned}
0 &< \phi(B_{k+1}) \\
&\le \phi(B_0)+\sigma(k+1)+\sum_{i=0}^{k_0-1}\ln\cos^2\theta_i+\sum_{i=k_0}^{k}\ln\cos^2\theta_i \\
&\le \phi(B_0)+\sigma(k+1)+\sum_{i=0}^{k_0-1}\ln\cos^2\theta_i+\sum_{i=k_0}^{k}(-2\sigma) \\
&= \phi(B_0)+\sum_{i=0}^{k_0-1}\ln\cos^2\theta_i+\sigma(k+1)-2\sigma(k-k_0+1) \\
&= \phi(B_0)+\sum_{i=0}^{k_0-1}\ln\cos^2\theta_i+2\sigma k_0-\sigma(k+1)\ \to\ -\infty,\ (k\to\infty),
\end{aligned}$$

矛盾. 这样便存在 $\{x_k\}$ 的无穷子列 $\{x_k\}_{k\in K}$ 和数 $\delta>0$, 使得对所有的 $k\in K$, 有 $\cos\theta_k\ge\delta$. 于是类似于定理 11 的证明过程, 可以推出 $\{\|g_k\|\}_{k\in K}\to 0$. 因 $f(x)$ 在水平集上是严格凸的, 其稳定点与全局极小点是一致的也是唯一的, 这样便可推得整个序列 $\{x_k\}$ 收敛到 $f(x)$ 的全局极小点 x^*. 证毕. □

下面我们给出拟牛顿法超线性收敛的一个充分必要条件.

定理 27 设 $f(x)$ 满足假设 1, $\{B_k\}$ 是非奇异的矩阵序列. 若迭代公式

$$x_{k+1}=x_k-B_k^{-1}g_k,\quad x_0\in\mathbb{R}^n \tag{5.44}$$

产生的无穷序列 $\{x_k\}$ 收敛于 $f(x)$ 的稳定点 x^*, 则 $\{x_k\}$ 超线性收敛到 x^* 的充分必要条件是

$$\lim_{k\to\infty}\frac{\|[B_k-G(x^*)]d_k\|}{\|d_k\|}=0, \tag{5.45}$$

其中 $d_k=x_{k+1}-x_k$, $G(x^*)=\nabla^2 f(x^*)$.

证 设牛顿步为 $d_k^N=-G_k^{-1}g_k$. 首先证明 (5.45) 等价于

$$\|d_k-d_k^N\|=o(\|d_k\|). \tag{5.46}$$

事实上, 若 (5.45) 式成立, 则

$$\begin{aligned}
\|d_k-d_k^N\| &= \|G_k^{-1}[G_kd_k+g_k]\| \\
&= \|G_k^{-1}[G_k-B_k]d_k\|\le\|G_k^{-1}\|\cdot\|[G_k-B_k]d_k\| \\
&\le C\big(\|[G_k-G(x^*)]d_k\|+\|[G(x^*)-B_k]d_k\|\big) \\
&= o(\|d_k\|).
\end{aligned}$$

反之, 若 (5.46) 式成立, 注意到 $\|G_k\|$ 是有界的, 故有

$$\|G_k(d_k - d_k^N)\| = o(\|d_k\|).$$

由 $G_k d_k^N = -g_k = B_k d_k$, 我们有

$$\|(G_k - B_k)d_k\| = o(\|d_k\|).$$

由上式及 $G(x)$ 的连续性即可推得 (5.45) 式成立.

注意到牛顿法的二阶收敛性结果

$$\|x_k + d_k^N - x^*\| \le C\|x_k - x^*\|^2,$$

我们有

$$\begin{aligned}
\|d_k\| - \|x_k - x^*\| &\le \|x_k + d_k - x^*\| \\
&\le \|x_k + d_k^N - x^*\| + \|d_k - d_k^N\| \\
&\le C\|x_k - x^*\|^2 + o(\|d_k\|).
\end{aligned}$$

由此可得 $\|d_k\| = O(\|x_k - x^*\|)$, 再代入上式即得

$$\|x_k + d_k - x^*\| = o(\|x_k - x^*\|).$$

至此已经完成了定理的证明. □

最后, 我们不加证明地列出 BFGS 算法的局部超线性收敛定理, 其详细的证明过程可参阅文献 [15].

定理 28 设 $f(x)$ 满足假设 1, x_0 和 B_0 为任意给定的初始点和初始正定对称矩阵. $\{x_k\}$ 是由 BFGS 算法产生的迭代序列且收敛于假设 1 (3) 中的 x^*. 那么, 若

$$\sum_{k=1}^{\infty} \|x_k - x^*\| < \infty,$$

则 $\{x_k\}$ 局部超线性收敛于 x^*.

习 题 5

1. 设用 DFP 算法求解问题的过程中, 有

$$H_k = \begin{pmatrix} 3 & 1 \\ 1 & 1 \end{pmatrix},\ s_k = \begin{pmatrix} 1 \\ 2 \end{pmatrix},\ y_k = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

计算 H_{k+1}.

2. 用 DFP 算法求解 $\min f(x) = x_1^2 + 3x_2^2$, 取初始点和初始矩阵为

$$x_0 = \begin{pmatrix} 1 \\ -1 \end{pmatrix},\ H_0 = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}.$$

3. 用 BFGS 算法求 $f(x) = x_1^2 + x_1x_2 + x_2^2$ 的极小点, 取初始点 $x_0 = (3, 2)^T$.

4. 分别利用 BFGS 算法和 DFP 算法的 Matlab 程序求解下列优化问题:

(1) $\min f(x) = x_1^2 + x_2^2 - 3x_1 - x_1x_2 + 3$, 取 $x_0 = (0, 0)^T$;

(2) $\min f(x) = 4(1 - x_1)^2 + 5(x_2 - x_1^2)^2$, 取 $x_0 = (2, 0)^T$.

5. 用 DFP 算法求解问题

$$\min f(x) = x_1^2 - x_1x_2 + x_2^2 + 2x_1 - 4x_2,$$

初始点取为 $x_0 = (2, 2)^T$, 初始矩阵取为 $H_0 = I$ (单位阵). 验证算法所生成的两个方向是关于

$$H = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$$

共轭的.

6. 分别利用 BFGS 算法和 DFP 算法的 Matlab 程序求 Powell 奇异函数的极小值:

$$\min f(x) = (x_1 + 10x_2)^2 + 5(x_3 - 10x_4)^2 + (x_2 - 2x_3)^2 + 10(x_1 - x_4)^2.$$

初始点取为 $x_0 = (3, -1, 0, 1)^T$.

7. 设 A 为 n 阶非奇异矩阵，$u, v \in \mathbb{R}^n$, 证明：$A + uv^T$可逆当且仅当$I + v^TA^{-1}u$ 可逆，且

$$(A + uv^T)^{-1} = A^{-1} - A^{-1}u(I + v^TA^{-1}u)^{-1}v^TA^{-1}.$$

8. 设 BFGS 的 Hessen 矩阵校正公式为:

$$B_{k+1} = B_k + \frac{y_ky_k^T}{y_k^Ts_k} - \frac{B_ks_ks_k^TB_k}{s_k^TB_ks_k}.$$

$H_k = B_k^{-1}$, $H_{k+1} = B_{k+1}^{-1}$ 且 $y_k^Ts_k > 0$, 试用习题 7 的求逆公式求 H_{k+1} 的表达式.

9. 在 DFP 校正公式

$$H_{k+1} = H_k + \frac{s_k s_k^T}{y_k^T s_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}$$

中, 记

$$A_k = \frac{s_k s_k^T}{y_k^T s_k}, \quad B_k = -\frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}.$$

假设H_1 对称正定, $\nabla f(x_k) \neq 0 (k = 1, 2, \cdots, n)$, 证明当 DFP 算法极小二次函数 $f(x) = \frac{1}{2}x^T G x + b^T x$ 时, 有

$$\sum_{k=1}^{n} A_k = G^{-1}, \quad \sum_{k=1}^{n} B_k = -H_1,$$

其中 G 对称正定.

第六章　信赖域方法

信赖域方法与线搜索技术一样, 也是优化算法中的一种保证全局收敛的重要技术. 它们的功能都是在优化算法中求出每次迭代的位移, 从而确定新的迭代点. 所不同的是: 线搜索技术是先产生位移方向 (亦称为搜索方向), 然后确定位移的长度 (亦称为搜索步长); 而信赖域技术则是直接确定位移, 产生新的迭代点.

信赖域方法的基本思想是: 首先给定一个所谓的“信赖域半径”作为位移长度的上界, 并以当前迭代点为中心以此“上界”为半径“画地为牢”确定一个称之为“信赖域”的闭球区域. 然后, 通过求解这个区域内的“信赖域子问题”(目标函数的二次近似模型) 的最优点来确定“候选位移”. 若候选位移能使目标函数值有充分的下降量, 则接受该候选位移作为新的位移, 并保持或扩大信赖域半径, 继续新的迭代; 否则, 说明二次模型与目标函数的近似度不够理想, 需要缩小信赖域半径, 再通过求解新的信赖域内的子问题得到新的候选位移. 如此重复下去, 直到满足迭代终止条件.

6.1　信赖域方法的基本结构

现在我们来讨论用信赖域方法求解无约束优化问题

$$\min_{x\in\mathbb{R}^n}\ f(x) \tag{6.1}$$

的基本算法结构. 设 x_k 是第 k 次迭代点. 记 $f_k=f(x_k)$, $g_k=\nabla f(x_k)$, B_k 是 Hesse 阵 $\nabla^2 f(x_k)$ 的第 k 次近似, 则第 k 次迭代步的信赖域子问题具有如下形式:

$$\begin{aligned} \min\ \ & q_k(d)=g_k^Td+\frac{1}{2}d^TB_kd, \\ \text{s.t.}\ \ & \|d\|\le\Delta_k, \end{aligned} \tag{6.2}$$

其中 Δ_k 是信赖域半径, $\|\cdot\|$ 是任一种向量范数, 通常取 2-范数或 ∞-范数. 设子问题 (6.2) 的最优解为 d_k, 定义 Δf_k 为 f 在第 k 步的实际下降量:

$$\Delta f_k = f_k - f(x_k + d_k),$$

Δq_k 为对应的预测下降量:

$$\Delta q_k = q_k(0) - q_k(d_k).$$

再定义它们的比值为

$$r_k = \frac{\Delta f_k}{\Delta q_k}. \tag{6.3}$$

一般地, 我们有 $\Delta q_k > 0$. 因此, 若 $r_k < 0$, 则 $\Delta f_k < 0$, $x_k + d_k$ 不能作为下一个迭代点, 需要缩小信赖域半径重新求解子问题. 若 r_k 比较接近 1, 说明二次模型与目标函数在信赖域范围内有很好的近似, 此时 $x_{k+1} := x_k + d_k$ 可以作为新的迭代点, 同时下一次迭代时可以增大信赖域半径. 对于其他情况, 信赖域半径可以保持不变. 下面给出求解无约束优化问题信赖域方法的一般框架.

算法 18 (信赖域方法)

步 0　选取初始参数 $0 \le \eta_1 < \eta_2 < 1$, $0 < \tau_1 < 1 < \tau_2$, $0 \le \varepsilon \ll 1$. $x_0 \in \mathbb{R}^n$. 取定 $\tilde{\Delta} > 0$ 为信赖域半径的上限, 初始信赖域半径 $\Delta_0 \in (0, \tilde{\Delta}]$. 令 $k := 0$.

步 1　计算 $g_k = \nabla f(x_k)$. 若 $\|g_k\| \le \varepsilon$, 停止迭代.

步 2　求解子问题 (6.2) 的解 d_k.

步 3　按 (6.3) 式计算 r_k 的值.

步 4　校正信赖域半径.

$$\Delta_{k+1} := \begin{cases} \tau_1 \Delta_k, & 若\ r_k \le \eta_1, \\ \Delta_k, & 若\ \eta_1 < r_k < \eta_2, \\ \min\{\tau_2 \Delta_k, \tilde{\Delta}\}, & 若\ r_k \ge \eta_2,\ \|d_k\| = \Delta_k. \end{cases} \tag{6.4}$$

步 5　若 $r_k > \eta_1$, 则令 $x_{k+1} := x_k + d_k$, 更新矩阵 B_k 到 B_{k+1}, 令 $k := k+1$, 转步 1. 否则 $x_{k+1} := x_k$, 令 $k := k+1$, 转步 2.

注　在算法 18 中一组推荐的参数值为

$$\eta_1 = 0.05, \quad \eta_2 = 0.75, \quad \tau_1 = 0.5, \quad \tau_2 = 2.0, \quad \Delta_0 = 1\ 或\ \Delta_0 = \frac{1}{10}\|g(x_0)\|.$$

在实际计算中可以上述参数进行调整, 以达到最佳计算效果.

由于子问题 (6.2) 的可行域是有界闭集, 因此算法 18 中步 2 的 d_k 存在, 即子问题 (6.2) 是可解的. 下面的定理说明 x_k 不是问题 (6.1) 的稳定点, 则预估下降量 $\Delta q_k > 0$. 因此算法是适定的.

定理 29 设 d_k 是子问题 (6.2) 的解. 若 $g_k = \nabla f(x_k) \neq 0$, 则

$$\Delta q_k = q_k(0) - q_k(d_k) > 0.$$

证 易知 0 是子问题 (6.2) 的可行点, 因此 $q_k(d_k) \leq q_k(0)$, 即 $\Delta q_k \geq 0$. 下面只需证明 $\Delta q_k \neq 0$. 如若不然, $\Delta q_k = q_k(0) - q_k(d_k) = 0$, 则 $q_k(d_k) = q_k(0)$. 故 0 是子问题 (6.2) 的最优解. 但 0 是可行域的内点, 故有 $\nabla q_k(0) = 0$, 即 $\nabla f(x_k) = 0$, 这与定理的假设矛盾. 证毕. □

利用上述定理可以证明由算法 18 产生的序列 $\{f(x_k)\}$ 是单调非增的. 我们有

推论 3 设 $\{x_k\}$ 是由算法 18 产生的迭代序列, 则序列 $\{f(x_k)\}$ 是单调非增的.

证 由算法结构可知, 对任意 $k \geq 0$, 若 $r_k \leq \eta_1$, 则 $x_{k+1} := x_k$, 此时有 $f(x_{k+1}) = f(x_k)$. 若 $r_k > \eta_1$, 由定理 29 以及 r_k 的定义可知

$$f(x_k) - f(x_{k+1}) = f(x_k) - f(x_k + d_k) = r_k \Delta q_k > 0,$$

即 $f(x_{k+1}) < f(x_k)$. 证毕. □

6.2　信赖域方法的收敛性

为了分析信赖域方法的收敛性, 我们首先在迭代点 x_k 处, 引入所谓的“柯西点”(Cauchy point) d_k^c 的定义:

$$d_k^c = -\tau_k \frac{\Delta_k}{\|g_k\|} g_k, \tag{6.5}$$

其中

$$\tau_k = \begin{cases} 1, & g_k^T B_k g_k \leq 0, \\ \min\left\{ \dfrac{\|g_k\|^3}{\Delta_k g_k^T B_k g_k}, 1 \right\}, & g_k^T B_k g_k > 0. \end{cases} \tag{6.6}$$

容易证明

$$\|d_k^c\| = \tau_k\Delta_k \le \Delta_k,$$

即柯西点是可行点, 且平行于 $f(x)$ 在 x_k 处的负梯度方向 (最速下降方向). 下面的引理说明柯西点 d_k^c 可以带来二次模型一定量的下降.

引理 7 由 (6.5) 定义的柯西点 d_k^c 满足

$$q_k(0) - q_k(d_k^c) \ge \frac{1}{2}\|g_k\| \min\left\{\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right\}. \tag{6.7}$$

证 (1) 若 $g_k^T B_k g_k \le 0$, 则由 (6.5)-(6.6) 有 $d_k^c = -\dfrac{\Delta_k}{\|g_k\|}g_k$. 此时有

$$\begin{aligned} q_k(0) - q_k(d_k^c) &= 0 - q_k\Big(-\frac{\Delta_k}{\|g_k\|}g_k\Big) \\ &= -g_k^T\Big(-\frac{\Delta_k}{\|g_k\|}g_k\Big) - \frac{1}{2}\Big(-\frac{\Delta_k}{\|g_k\|}g_k\Big)^T B_k\Big(-\frac{\Delta_k}{\|g_k\|}g_k\Big) \\ &= \frac{\Delta_k}{\|g_k\|}\|g_k\|^2 - \frac{1}{2}\frac{\Delta_k^2}{\|g_k\|^2}g_k^T B_k g_k \\ &\ge \frac{\Delta_k}{\|g_k\|}\|g_k\|^2 = \Delta_k\|g_k\| \quad (\text{注意到 } -g_k^T B_k g_k \ge 0) \\ &\ge \frac{1}{2}\|g_k\| \min\left\{\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right\}. \end{aligned}$$

(2) 若 $g_k^T B_k g_k > 0$ 且 $\dfrac{\|g_k\|^3}{\Delta_k g_k^T B_k g_k} \le 1$, 则 $d_k^c = -\dfrac{\|g_k\|^2}{g_k^T B_k g_k}g_k$, 从而有

$$\begin{aligned} q_k(0) - q_k(d_k^c) &= 0 - q_k\Big(-\frac{\|g_k\|^2}{g_k^T B_k g_k}g_k\Big) \\ &= -g_k^T\Big(-\frac{\|g_k\|^2}{g_k^T B_k g_k}g_k\Big) - \frac{1}{2}\Big(-\frac{\|g_k\|^2}{g_k^T B_k g_k}g_k\Big)^T B_k\Big(-\frac{\|g_k\|^2}{g_k^T B_k g_k}g_k\Big) \\ &= \frac{1}{2}\frac{\|g_k\|^4}{g_k^T B_k g_k} \ge \frac{1}{2}\frac{\|g_k\|^2}{\|B_k\|} \ge \frac{1}{2}\|g_k\| \min\left\{\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right\}. \end{aligned}$$

若 $g_k^TB_kg_k>0$ 且 $\dfrac{\|g_k\|^3}{\Delta_k g_k^TB_kg_k}>1$, 则 $d_k^c=-\dfrac{\Delta_k}{\|g_k\|}g_k$ 及 $\|g_k\|^3>\Delta_k g_k^TB_kg_k$, 且有

$$\begin{aligned}
q_k(0)-q_k(d_k^c) &= 0-q_k\Big(-\frac{\Delta_k}{\|g_k\|}g_k\Big)\\
&= -g_k^T\Big(-\frac{\Delta_k}{\|g_k\|}g_k\Big)-\frac{1}{2}\Big(-\frac{\Delta_k}{\|g_k\|}g_k\Big)^TB_k\Big(-\frac{\Delta_k}{\|g_k\|}g_k\Big)\\
&= \frac{\Delta_k}{\|g_k\|}\|g_k\|^2-\frac{1}{2}\frac{\Delta_k^2}{\|g_k\|^2}g_k^TB_kg_k\\
&\ge \frac{1}{2}\Delta_k\|g_k\| \ \ge\ \frac{1}{2}\|g_k\|\min\Big\{\Delta_k,\frac{\|g_k\|}{\|B_k\|}\Big\}.
\end{aligned}$$

这表明在各种情况下, (6.7) 都成立. 证毕. □

推论 4 设 d_k 是信赖域子问题 (6.2) 的解, 则有

$$q_k(0)-q_k(d_k)\ge\frac{1}{2}\|g_k\|\min\Big\{\Delta_k,\frac{\|g_k\|}{\|B_k\|}\Big\}. \tag{6.8}$$

证 由于 $q_k(d_k)\le q_k(d_k^c)$, 由引理 7 立即可得结论. 证毕. □

下面给出算法 18 的全局收敛性定理.

定理 30 假设在算法 18 中取 $\varepsilon=0$, 函数 $f(x)$ 有下界, 且对任意的 $x_0\in\mathbb{R}^n$, f 在水平集 $L(x_0)=\{x\in\mathbb{R}^n|\,f(x)\le f(x_0)\}$ 上连续可微. 又设 d_k 是子问题 (6.2) 的解, 且矩阵序列 $\{B_k\}$ 一致有界, 即存在常数 $M>0$ 使对任意的 k 满足 $\|B_k\|\le M$. 那么若 $g_k\ne 0$, 则必有

$$\liminf_{k\to\infty}\|g_k\|=0.$$

证 不失一般性, 设 $g_k\ne0$. 则由 r_k 的定义有

$$\begin{aligned}
|r_k-1| &= \left|\frac{[f_k-f(x_k+d_k)]-[q_k(0)-q_k(d_k)]}{q_k(0)-q_k(d_k)}\right|\\
&= \left|\frac{f(x_k+d_k)-f_k-q_k(d_k)}{q_k(0)-q_k(d_k)}\right|.
\end{aligned}\tag{6.9}$$

注意到由泰勒公式有

$$f(x_k+d_k)=f_k+g_k^Td_k+\int_0^1 d_k^T[g(x_k+\tau d_k)-g_k]\mathrm{d}\tau.$$

因此当 $\Delta_k > 0$ 充分小时, 可得

$$\begin{aligned}|f(x_k+d_k)-f_k-q_k(d_k)| &= \left|\frac{1}{2}d_k^TB_kd_k-\int_0^1 d_k^T[g(x_k+\tau d_k)-g_k]\mathrm{d}\tau\right| \\ &\le \frac{1}{2}M\|d_k\|^2+o(\|d_k\|). \qquad (6.10)\end{aligned}$$

以下用反证法证明定理的结论. 设存在 $\varepsilon_0 > 0$ 使得 $\|g_k\| \ge \varepsilon_0$. 于是由 (6.8)-(6.10) 得

$$|r_k-1| \le \frac{\frac{1}{2}M\|d_k\|^2+o(\|d_k\|)}{\frac{1}{2}\|g_k\|\min\left\{\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right\}} \le \frac{M\Delta_k^2+o(\Delta_k)}{\varepsilon_0\min\left\{\Delta_k, \frac{\varepsilon_0}{M}\right\}} = O(\Delta_k).$$

上式表明存在充分小的 $\bar{\Delta} > 0$, 使得对任意满足 $\Delta_k \le \bar{\Delta}$ 的 k, 都有

$$|r_k-1| \le 1-\eta_2,$$

即 $r_k \ge \eta_2$. 根据算法 18 有 $\Delta_{k+1} \ge \Delta_k$. 故存在正整数 k_0 和常数 $\gamma > 0$, 对任意满足 $\Delta_k \le \bar{\Delta}$ 的 $k \ge k_0$ 有

$$\Delta_k \ge \gamma\bar{\Delta}. \qquad (6.11)$$

另一方面, 假定存在无穷多个 k 满足 $r_k \ge \eta_1$. 则由 r_k 的定义和 (6.8), 对任意的 $k \ge k_0$ 有

$$f_k-f_{k+1} \ge \eta_1[q_k(0)-q_k(d_k)] \ge \frac{\eta_1}{2}\varepsilon_0\min\left\{\Delta_k, \frac{\varepsilon_0}{M}\right\}.$$

因 $f(x)$ 有下界, 上式意味着 $\Delta_k \to 0\,(k\to\infty)$, 这与 (6.11) 矛盾.

再假定对于充分大的 k, 都有 $r_k < \eta_1$ 成立. 此时 Δ_k 将以 $\tau_1(<1)$ 的比例收缩, 同样有 $\Delta_k \to 0\,(k\to\infty)$, 也与 (6.11) 矛盾. 因此前面的假设 $\|g_k\| \ge \varepsilon_0$ 不成立, 从而定理的结论成立. 证毕. □

6.3　信赖域子问题的求解

信赖域方法中子问题的求解是算法实现的关键. 信赖域子问题 (6.2) 是一个目标函数为二次函数的约束优化问题. 已经建立了求解该子问题的如果数值方法, 如折线法, 截断共轭梯度法以及特征值分解法等. 本书介绍一种新的求解方法.

首先我们引述下面的定理.

定理 31　d 是子问题

$$\begin{cases} \min\ q_k(d) = g_k^T d + \dfrac{1}{2} d^T B_k d, \\ \text{s.t.}\ \ \|d\|_2 \le \Delta_k \end{cases} \tag{6.12}$$

的解当且仅当

$$\begin{cases} (B_k + \lambda I)d - g_k = 0, \\ \lambda \ge 0,\ \ \Delta_k^2 - \|d\|_2^2 \ge 0,\ \ \lambda(\Delta_k^2 - \|d\|_2^2) = 0, \end{cases} \tag{6.13}$$

而且 $B_k + \lambda I$ 是半定矩阵.

定义一个函数 $\mathbb{R}_+ \times \mathbb{R}^2 \to \mathbb{R}$:

$$\phi(\mu, a, b) = a + b - \sqrt{(a-b)^2 + 4\mu^2}.$$

则不难推出该函数具有如下性质

$$\phi(0, a, b) = 0 \ \Leftrightarrow\ a \ge 0,\ \ b \ge 0,\ \ ab = 0.$$

令 $z = (\mu, \lambda, d) \in \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R}^n$. 于是, 问题 (6.13) 等价于

$$H(z) := H(\mu, \lambda, d) = \begin{pmatrix} \mu \\ \phi\big(\mu, \lambda, \Delta_k^2 - \|d\|_2^2\big) \\ (B_k + \lambda I)d - g_k \end{pmatrix} = 0, \tag{6.14}$$

其中

$$\phi(\mu, \lambda, \Delta_k^2 - \|d\|_2^2) = \lambda - \|d\|_2^2 + \Delta_k^2 - \sqrt{(\lambda + \|d\|_2^2 - \Delta_k^2)^2 + 4\mu^2}. \tag{6.15}$$

不难证明, 当 $\mu > 0$ 时, 由 (6.15) 定义的函数 ϕ 是连续可微的, 且有

$$\begin{aligned}
\phi'_\mu(\mu, \lambda, \Delta_k^2 - \|d\|_2^2) &= -\frac{4\mu}{\sqrt{(\lambda + \|d\|_2^2 - \Delta_k^2)^2 + 4\mu^2}}, \\
\phi'_\lambda(\mu, \lambda, \Delta_k^2 - \|d\|_2^2) &= 1 - \frac{\lambda + \|d\|_2^2 - \Delta_k^2}{\sqrt{(\lambda + \|d\|_2^2 - \Delta_k^2)^2 + 4\mu^2}} := 1 - \theta_k, \\
\phi'_d(\mu, \lambda, \Delta_k^2 - \|d\|_2^2) &= -2d^T\left(1 + \frac{\lambda + \|d\|_2^2 - \Delta_k^2}{\sqrt{(\lambda + \|d\|_2^2 - \Delta_k^2)^2 + 4\mu^2}}\right) := -2(1 + \theta_k)d^T.
\end{aligned}$$

故当 $\mu > 0$ 时, $H(\cdot)$ 是连续可微的, 且其 Jacobi 矩阵为

$$H'(z) = \begin{pmatrix} 1 & 0 & 0 \\ \phi'_\mu & \phi'_\lambda & -2(1+\theta_k)d^T \\ 0 & d & B_k + \lambda I \end{pmatrix}. \tag{6.16}$$

不难证明, 若 B_k 对称正定, 则对任意的 $z = (\mu, \lambda, d) \in \mathbb{R}_{++} \times \mathbb{R}_+ \times \mathbb{R}^n$, Jacobi 矩阵 $H'(z)$ 是非奇异的.

给定参数 $\gamma \in (0,1)$, 定义非负函数

$$\beta(z) = \gamma \|H(z)\| \min\{1, \|H(z)\|\}. \tag{6.17}$$

算法 19 (求解子问题的光滑牛顿法)

步 0 选取 δ, $\sigma \in (0,1)$, $\mu_0 > 0$, $\lambda_0 \geq 0$. $d_0 \in R^n$, 置 $z_0 = (\mu_0, \lambda_0, d_0)$, $\bar{z} = (\mu_0, 0, 0)$. 选取 $\gamma \in (0,1)$ 使 $\gamma\mu_0 < 1$ 及 $\gamma\|H(z_0)\| < 1$. 令 $j := 0$.

步 1 如果 $\|H(z_j)\| = 0$, 算法终止; 否则, 计算 $\beta_j = \beta(z_j)$.

步 2 求解下列方程组得解 $\Delta z_j = (\Delta\mu_j, \Delta\lambda_j, \Delta d_j)$,

$$H(z_j) + H'(z_j)\Delta z_j = \beta_j \bar{z}. \tag{6.18}$$

步 3 设 m_j 为满足下式的最小非负整数:

$$\|H(z_j + \delta^{m_j}\Delta z_j)\| \leq [1 - \sigma(1 - \beta\mu_0)\delta^{m_j}]\|H(z_j)\|. \tag{6.19}$$

令 $\alpha_j := \delta^{m_j}$, $z_{j+1} = z_j + \alpha_j \Delta z_j$.

步 4 令 $j := j + 1$, 转步 1.

算法 19 的适定性及收敛性定理的证明类似于文献 [1], 此处省略不证.

下面我们给出算法 19 的 Matlab 程序.

程序 12 利用光滑牛顿法求解信赖域子问题, 一般适用于 (近似) Hesse 阵正定的情形.

```
function  [d,val,lam,k]=trustq(gk,Bk,dta)
% 功能: 求解信赖域子问题:  min qk(d)=gk'*d+0.5*d'*Bk*d, s.t. ||d||<=delta
%输入:  gk是xk处的梯度, Bk是第k次近似Hesse阵, dta是当前信赖域半径
%输出:  d, val分别是子问题的最优点和最优值, lam是乘子值, k是迭代次数.
n=length(gk);   gamma=0.05;
```

```
epsilon=1.0e-6;  rho=0.6;  sigma=0.2;
mu0=0.05;  lam0=0.05;
d0=ones(n,1);  u0=[mu0,zeros(1,n+1)]';
z0=[mu0,lam0,d0']';
k=0;   %k为迭代次数
z=z0; mu=mu0; lam=lam0; d=d0;
while (k<=150)
     dh=dah(mu,lam,d,gk,Bk,dta);
     if(norm(dh)<epsilon)
         break;
     end
     A=JacobiH(mu,lam,d,Bk,dta);
     b=beta(mu,lam,d,gk,Bk,dta,gamma)*u0-dh;
     B=inv(A);   dz=B*b;
     dmu=dz(1); dlam=dz(2); dd=dz(3:n+2);
     m=0;  mk=0;
     while (m<20)
         dhnew=dah(mu+rho^m*dmu,lam+rho^m*dlam,d+rho^m*dd,gk,Bk,dta);
         if(norm(dhnew)<=(1-sigma*(1-gamma*mu0)*rho^m)*dh)
             mk=m;
             break;
         end
         m=m+1;
     end
     alpha=rho^mk;
     mu=mu+alpha*dmu;
     lam=lam+alpha*dlam;
     d=d+alpha*dd;
     k=k+1;
end
val=gk'*d+0.5*d'*Bk*d;
%%%%%%%%%%%%%%%%%%%%%%%%%%
function p=phi(mu,a,b)
p=a+b-sqrt((a-b)^2+4*mu);
%%%%%%%%%%%%%%%%%%%%%%%%%%
function dh=dah(mu,lam,d,gk,Bk,dta)
n=length(d);
```

```
dh(1)=mu;  dh(2)=phi(mu,lam, dta^2-norm(d)^2);
mh=(Bk+lam*eye(n))*d+gk;
for(i=1:n)
    dh(2+i)=mh(i);
end
dh=dh(:);
%%%%%%%%%%%%%%%%%%%%%%%%%%
function bet=beta(mu,lam,d,gk,Bk,dta,gamma)
dh=dah(mu,lam,d,gk,Bk,dta);
bet=gamma*norm(dh)*min(1,norm(dh));
%%%%%%%%%%%%%%%%%%%%%%%%%%
function A=JacobiH(mu,lam,d,Bk,dta)
n=length(d);
A=zeros(n+2,n+2);
pmu=-4*mu/sqrt((lam+norm(d)^2-dta^2)^2+4*mu^2);
thetak=(lam+norm(d)^2-dta^2)/sqrt((lam+norm(d)^2-dta^2)^2+4*mu^2);
A=[1,            0,           zeros(1,n);
   pmu,       1-thetak,  -2*(1+thetak)*d';
   zeros(n,1),    d,         Bk+lam*eye(n)];
```

我们利用上面的程序求解一个信赖域子问题.

例 19　求下面信赖域子问题的最优解

$$\min_{\|d\|_2\le\Delta} q_k(d) = g_k^T d + \frac{1}{2} d^T B_k d,$$

其中

$$g_k = \begin{pmatrix} 400 \\ -200 \end{pmatrix}, \quad B_k = \begin{pmatrix} 1202 & -400 \\ -400 & 200 \end{pmatrix}, \quad \Delta = 5.$$

解　在 Matlab 命令窗口依次输入下列命令

```
gk=[400 -200]';
Bk=[1202 -400; -400 200];
dta=5;
[d,val,lam,k]=trustq(gk,Bk,dta)
```

得到

```
d =
   0.0000
   1.0000
val =
   -100.0000
lam =
   4.1000e-007
k =
   5
```

即子问题的最优解为 $d = (0, 1)^T$, 最优值为 $q_k(d) = -100$, 迭代 5 次.

6.4　信赖域方法的 Matlab 程序

本节给出一个牛顿型信赖域方法的 Matlab 程序, 在某种意义上该程序是通用的.

程序 13 (**求解例 20 的信赖域方法 Matlab 程序**)

```
function [xk,val,k]=trustm(x0)
%功能: 牛顿型信赖域方法求解无约束优化问题 min f(x)
%输入: x0是初始迭代点
%输出: xk是近似极小点, val是近似极小值, k是迭代次数
n=length(x0);  x=x0; dta=1;
eta1=0.1; eta2=0.75;  dtabar=2.0;
tau1=0.5; tau2=2.0; epsilon=1e-6;
k=0; Bk=Hess(x);
while(k<50)
    gk=gfun(x);
    if(norm(gk)<epsilon)
        break;
    end
    % 调用子程序trustq
    [d,val,lam,ik]=trustq(gk,Bk,dta);
    deltaq=-qk(x,d);
    deltaf=fun(x)-fun(x+d);
    rk=deltaf/deltaq;
    if(rk<=eta1)
```

```
        dta=tau1*dta;
    else if (rk>=eta2&norm(d)==dta)
            dta=min(tau2*dta,dtabar);
        else
            dta=dta;
        end
    end
    if(rk>eta1)
       x=x+d;
       Bk=Hess(x);
    end
    k=k+1;
end
xk=x;
val=fun(xk);
```

例 20 利用信赖域方法的程序 13 求解无约束优化问题

$$\min_{x\in\mathbb{R}^n} f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2. \tag{6.20}$$

该问题有精确解 $x^* = (1,1)^T$, $f(x^*) = 0$. 所谓牛顿型信赖域方法, 是指信赖域子问题中的矩阵 B_k 取为目标函数的 Hesse 阵 $G_k = \nabla^2 f(x)$.

解 我们需要先编制目标函数和它的梯度及 Hesse 阵的三个 M 文件:

目标函数文件 fun.m

```
function f=fun(x)
f=100*(x(1)^2-x(2))^2+(x(1)-1)^2;
```

目标函数的梯度文件 gfun.m

```
function gf=gfun(x)
gf=[400*x(1)*(x(1)^2-x(2))+2*(x(1)-1), -200*(x(1)^2-x(2))]';
```

目标函数的 Hesse 阵文件 Hess.m

```
function He=Hess(x)
He=[1200*x(1)^2-400*x(2)+2, -400*x(1); -400*x(1), 200];
```

信赖域子问题目标函数文件 qk.m

```
function qd=qk(x,d)
gk=gfun(x);  Bk=Hess(x);
qd=gk'*d+0.5*d'*Bk*d;
```

我们利用程序 13, 终止准则取为 $\|\nabla f(x_k)\| \le 10^{-6}$. 取不同的初始点, 数值结果如下表.

表 3.2 信赖域方法的数值结果.

初始点 (x_0)	迭代次数 (k)	目标函数值 ($f(x_k)$)
$(0.0, 0.0)^T$	19	9.9111×10^{-22}
$(0.5, 0.5)^T$	17	5.7509×10^{-23}
$(1.0, 2.0)^T$	35	1.1261×10^{-23}
$(2.0, 1.0)^T$	30	1.0828×10^{-17}
$(1.0, -1.0)^T$	18	1.5543×10^{-21}
$(-1.0, 1.0)^T$	36	1.3927×10^{-16}

说明　对于其他不同目标函数的无约束最优化问题的求解, 只需根据具体函数表达式修改目标函数、梯度和 Hesse 阵三个子函数程序 (fun, gfun, Hess) 即可.

习　题　6

1. 设矩阵 $B_k \in \mathbb{R}^{n\times n}$ 对称正定. d_k 是子问题 $\min q_k(d) = \frac{1}{2}d^T B_k d + \nabla f(x_k)^T d + f(x_k)$ 的解, 证明: d_k 是函数 f 在 x_k 处的下降方向.

2. 设函数 $q_k(d) = \frac{1}{2}d^T B_k d + \nabla f(x_k)^T d + f(x_k)$, 证明:

$$\tau_k = \begin{cases} 1, & \text{如果 } \nabla f(x_k)^T B_k \nabla f(x_k) \le 0, \\ \min\left\{1, \dfrac{\|\nabla f(x_k)\|^3}{\Delta_k \nabla f(x_k)^T B_k \nabla f(x_k)}\right\}, & \text{其它} \end{cases}$$

是沿搜索方向

$$\frac{\Delta_k}{\|\nabla f(x_k)\|}\nabla f(x_k)$$

的 $q_k(d)$ 的极小点.

3. 用信赖域方法的 Matlab 程序求最优化问题 $\min f(x) = 10(x_2 - x_1)^2 + (1 - x_1)^2$ 的最优解, 取初始点 $x_0 = (0,0)^T$, 终止准则值 $\varepsilon = 10^{-6}$.

4. 若对称矩阵 $B \in \mathbb{R}^{n\times n}$ 可分解为 $B = Q\Lambda Q^T$, 其中 $Q = (q_1, q_2, \cdots, q_n)$ 为正交阵, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$. 证明: 由

$$\begin{cases} (B + \lambda I)d = -g, \\ \|d\| = \Delta \end{cases}$$

所确定的解可表示为:

$$d(\lambda) = -\sum_{i=1}^{n} \frac{q_i^T g}{\lambda_i + \lambda} q_i.$$

进一步, 证明:

$$\frac{d}{d\lambda}(\|d(\lambda)\|^2) = -2\sum_{i=1}^{n} \frac{(q_i^T g)^2}{(\lambda_i + \lambda)^3}.$$

5. 设 B 是一个 n 阶对称矩阵, 证明存在 $\lambda \geq 0$, 使得 $B + \lambda I$ 是正定矩阵.

6. 设 $\tau \in [0,1]$ 满足 $\|(1-\tau)s_k^u + \tau s_k^N\| = \Delta_k$, 证明: $m_k(s_k) \leq m_k(s_k^c)$, 其中 $m_k(s) = f(x_k) + g_k^T s + \frac{1}{2}s^T B_k s$, $g_k = \nabla f(x_k)$,

$$s_k^c = \begin{cases} -\dfrac{\Delta_k}{\|g_k\|} g_k, & \text{如果 } g_k^T B_k g_k \leq 0, \\ -\min\left\{\dfrac{\Delta_k}{\|g_k\|}, \dfrac{\|g_k\|^2}{g_k^T B_k g_k}\right\} g_k, & \text{否则}, \end{cases}$$

$$s_k = \begin{cases} -\dfrac{\Delta_k}{\|g_k\|} g_k, & \text{若 } \|s_k^u\| \geq \Delta_k, \\ -B_k^{-1} g_k, & \text{若 } \|s_k^u\| < \Delta_k \text{ 且 } \|s_k^N\| \leq \Delta_k, \\ s_k^u + \tau(s_k^N - s_k^u), & \text{其它}. \end{cases}$$

7. 设 $s_k = \mathrm{argmin}\{m_k(s) : \|s\| \leq \Delta_k,\ s \in \mathrm{span}[g_k, B_k^{-1} g_k]\}$, 其中

$$m_k(s) = f(x_k) + g_k^T s + \frac{1}{2}s^T B_k s.$$

若 B_k 是正定的, 试确定 s_k 的具体表达式.

8. 利用光滑牛顿法的 Matlab 程序求解下列信赖域子问题, 分别取 $\Delta = 1,\ 2,\ 5$.

$$(1)\quad \begin{cases} \min\ q(x) = 2x_1^2 - 4x_1x_2 + 4x_2^2 - 6x_1 - 3x_2, \\ \text{s.t.}\quad \|x\| \leq \Delta \end{cases}.$$

$$(2)\quad \begin{cases} \min\ q(x) = \dfrac{1}{2}x^T A x + b^T x, \\ \text{s.t.}\quad \|x\| \leq \Delta, \end{cases}$$

其中

$$A=\begin{pmatrix} 3 & -1 & 2 \\ -1 & 2 & 0 \\ 2 & 0 & 4 \end{pmatrix},\quad b=\begin{pmatrix} 1 \\ -3 \\ -2 \end{pmatrix},\quad x=\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

第七章　非线性最小二乘问题

非线性最小二乘问题是科学与工程计算中十分常见的一类问题, 并在经济学等领域有广泛的应用背景. 不但如此, 约束优化问题还可以通过 KKT 条件与非线性方程组建立起重要的关系. 本章, 我们主要讨论非线性最小二乘问题的一些求解算法及其收敛性质.

7.1　Gauss-Newton 法

非线性最小二乘问题是求向量任 $x \in R^n$ 使 $\|F(x)\|^2$ 最小, 其中, 映射 $F: R^n \to R^m$ 是连续可微函数. 非线性最小二乘问题在工程设计、财政金融等方面的实际问题中有着广泛的应用.

记 $F(x) = (F_1(x), F_2(x), \cdots, F_m(x))^T$. 则非线性最小乘问题可以表示为

$$\min_{x \in R^n} f(x) = \frac{1}{2}\|F(x)\|^2 = \frac{1}{2}\sum_{i=1}^{m} F_i^2(x). \tag{7.1}$$

显然, 该问题本身就是一个无约束优化问题, 因此可以套用无约束优化问题的数值方法如牛顿法、拟牛顿法等方法求解. 基于问题 (7.1) 的特殊性, 我们在这些优化算法的基础上, 建立更适合本类问题的求解算法.

对于问题 (7.1), 目标函数 f 的梯度和 Hesse 阵分别为:

$$g(x) \triangleq \nabla f(x) = \nabla\Big(\frac{1}{2}\|F(x)\|^2\Big) = J(x)^T F(x) = \sum_{i=1}^{m} F_i(x)\nabla F_i(x),$$

$$\begin{aligned} G(x) \triangleq \nabla^2 f(x) &= \sum_{i=1}^m \nabla F_i(x)(\nabla F_i(x))^T + \sum_{i=1}^m F_i(x)\nabla^2 F_i(x) \\ &= J(x)^T J(x) + \sum_{i=1}^m F_i(x)\nabla^2 F_i(x) \\ &\triangleq J(x)^T J(x) + S(x), \end{aligned}$$

其中

$$J(x) = F'(x) = (\nabla F_1(x), \cdots, \nabla F_m(x))^T, \quad S(x) = \sum_{i=1}^m F_i(x)\nabla^2 F_i(x).$$

利用牛顿型迭代算法, 我们便得到求解非线性最小二乘问题的迭代算法:

$$x_{k+1} = x_k - \left(J_k^T J_k + S_k\right)^{-1} J_k^T F(x_k).$$

在标准假设下, 容易得到该算法的收敛性质. 缺点是 $S(x)$ 中 $\nabla^2 F_i(x)$ 的计算量较大. 如果忽略这一项, 便得到求解非线性最小二乘问题的 Gauss-Newton 迭代算法:

$$x_{k+1} = x_k + d_k^{GN},$$

其中

$$d_k^{GN} = -\left[J_k^T J_k\right]^{-1} J_k^T F(x_k)$$

称为 Gauss-Newton 方向. 容易验证 d_k^{GN} 是优化问题

$$\min_{d\in R^n} \frac{1}{2}\|F(x_k) + J_k d\|^2$$

的最优解. 若向量函数 $F(x)$ 的 Jacobian 矩阵是列满秩的, 则可以保证 Gauss-Newton 方向是下降方向. 如同牛顿法一样, 若采取单位步长, 算法的收敛性难以保证. 但如果在算法中引入线搜索步长规则, 则可以得到如下的收敛性定理.

定理 32 设水平集 $\mathcal{L}(x_0)$ 有界, $J(x) = F'(x)$ 在 $\mathcal{L}(x_0)$ 上 Lipschitz 连续且满足一致性条件

$$\|J(x)y\| \geq \alpha\|y\|, \quad \forall\, y \in R^n, \tag{7.2}$$

其中, $\alpha > 0$ 为一常数. 则在 Wolfe 步长规则下

$$\begin{cases} f(x_k + \alpha_k d_k) \leq f_k + \sigma_1 \alpha_k g_k^T d_k, \\ g(x_k + \alpha_k d_k)^T d_k \geq \sigma_2 g_k^T d_k, \end{cases} \tag{7.3}$$

其中 $0 < \sigma_1 < \sigma_2 < 1$. Gauss-Newton 算法产生的迭代点列 $\{x_k\}$ 收敛到 (7.1) 的一个稳定点. 即

$$\lim_{k\to\infty} J(x_k)^T F(x_k) = 0.$$

证明 由 $J(x)$ 在 $\mathcal{L}(x_0)$ 上 Lipschitz 连续可知 $J(x)$ 连续. 由于水平集 $\mathcal{L}(x_0)$ 有界, 故存在 $\beta > 0$ 使得对任意 $x \in \mathcal{L}(x_0)$, $\|J(x)\| \le \beta$ 成立. 记 θ_k 为 Gauss-Newton 方向 d_k^{GN} 与负梯度方向 $-g_k$ 的夹角. 利用一致性条件 (7.2), 我们有

$$\begin{aligned}\cos\theta_k &= -\frac{g_k^T d_k^{GN}}{\|g_k\|\|d_k^{GN}\|} = -\frac{F_k^T J_k d_k^{GN}}{\|d_k^{GN}\|\|J_k^T F_k\|}\\ &= \frac{\|J_k d_k^{GN}\|^2}{\|d_k^{GN}\|\|J_k^T J_k d_k^{GN}\|} \ge \frac{\alpha^2\|d_k^{GN}\|^2}{\beta^2\|d_k^{GN}\|^2} = \frac{\alpha^2}{\beta^2} > 0.\end{aligned}$$

由于 $g(x)$ 在 $\mathcal{L}(x_0)$ 上 Lipschitz 连续, 则由 (7.3) 的第二式得

$$(\sigma_2 - 1)g_k^T d_k \le [g(x_k + \alpha_k d_k) - g_k]^T d_k \le \alpha_k L\|d_k\|^2.$$

故

$$\alpha_k \ge \frac{\sigma_2 - 1}{L}\frac{g_k^T d_k}{\|d_k\|^2}.$$

将其代入 (7.3) 的第一式得

$$\begin{aligned}f_k - f_{k+1} &\ge -\sigma_1\alpha_k g_k^T d_k \ge \sigma_1\frac{1-\sigma_2}{L}\frac{(g_k^T d_k)^2}{\|d_k\|^2}\\ &= \sigma_1\frac{1-\sigma_2}{L}\|g_k\|^2\cos^2\theta_k.\end{aligned}$$

两边对 k 求级数, 利用 $\{f_k\}$ 单调不增有下界, 得到

$$\sum_{k=1}^{\infty}\|g_k\|^2\cos^2\theta_k < \infty.$$

由此可得

$$\lim_{k\to\infty} g_k = \lim_{k\to\infty} J(x_k)^T F(x_k) = 0.$$

证毕. □

定理 33 设单位步长的 Gauss-Newton 算法产生的迭代点列 $\{x_k\}$ 收敛到 (7.1) 的局部极小点 x^*, 而且 $J(x^*)^T J(x^*)$ 正定. 则当 $J(x)^T J(x)$, $S(x)$, $[J(x)^T J(x)]^{-1}$ 在 x^* 的邻域内 Lipschitz 连续时, 对充分大的 k , 有

$$\|x_{k+1} - x_k\| \le \|[J(x^*)^T J(x^*)]^{-1}\|\|S(x^*)\|\|x_k - x^*\| + O(\|x_k - x^*\|^2).$$

证明 由于 $J(x)^TJ(x)$, $S(x)$, $[J(x)^TJ(x)]^{-1}$ 在 x^* 的邻域内 Lipschitz 连续, 故存在 $\delta > 0$ 及正数 α, β, γ 使得对任意 $x, y \in N(x^*, \delta)$ 有,

$$\begin{cases} \|S(x) - S(y)\| \le \alpha\|x - y\| \\ \|J(x)^TJ(x) - J(y)^TJ(y)\| \le \beta\|x - y\|, \\ \|[J(x)^TJ(x)]^{-1} - [J(y)^TJ(y)]^{-1}\| \le \gamma\|x - y\|. \end{cases} \tag{7.4}$$

由于 $f(x)$ 二阶连续可微, $G(x) = J(x)^TJ(x) + S(x)$ 在 $N(x^*, \delta)$ 上 Lipschitz 连续, 故对充分大的 k 和模充分小的 $h \in R^n$, 有 $x_k + h \in N(x^*, \delta)$, 且

$$g(x_k + h) = g(x_k) + G(x_k)h + O(\|h\|^2). \tag{7.5}$$

由于 $x_k \to x^*$, 对充分大的 k , 有 $x_k, x_{k+1} \in N(x^*, \delta)$. 令 $e_k = x_k - x^*$, $h_k = x_{k+1} - x_k$, 则

$$g(x^*) = g(x_k - e_k) = 0.$$

利用 (7.5),

$$g(x_k) - G(x_k)e_k + O(\|e_k\|^2) = 0.$$

即

$$J_k^TF_k - (J_k^TJ_k + S_k)e_k + O(\|e_k\|^2) = 0.$$

注意到 $J_k^TF_k = -(J_k^TJ_k)(x_{k+1} - x_k) = -J_k^TJ_kh_k$. 两边同乘 $(J_k^TJ_k)^{-1}$,

$$-h_k - e_k - (J_k^TJ_k)^{-1}S_ke_k + (J_k^TJ_k)^{-1}O(\|e_k\|^2) = 0.$$

所以

$$x_{k+1} - x^* = h_k + e_k = -(J_k^TJ_k)^{-1}S_ke_k + (J_k^TJ_k)^{-1}O(\|e_k\|^2).$$

两边取 2 范数,

$$\|x_{k+1} - x^*\| \le \|(J_k^TJ_k)^{-1}S_k\|\|e_k\| + \|(J_k^TJ_k)^{-1}\| \cdot O(\|e_k\|^2).$$

由于 $[J(x)^TJ(x)]^{-1}$ 在 x^* 处连续, 故在 k 充分大时,

$$\|(J_k^TJ_k)^{-1}\| \le 2\|[J(x^*)^TJ(x^*)]^{-1}\|. \tag{7.6}$$

从而

$$\|x_{k+1} - x^*\| \le \|(J_k^TJ_k)^{-1}S_k\|\|x_k - x^*\| + O(\|x_k - x^*\|^2). \tag{7.7}$$

由 (7.4), (7.6) 得

$$\begin{aligned}
&\|(J_k^TJ_k)^{-1}S_k - [J(x^*)^TJ(x^*)]^{-1}S(x^*)\| \\
&\le \|(J_k^TJ_k)^{-1}\|\|S_k - S(x^*)\| \\
&+ \|(J_k^TJ_k)^{-1} - [J(x^*)^TJ(x^*)]^{-1}\|\|S(x^*)\| \\
&\le 2\alpha\|[J(x^*)^TJ(x^*)]^{-1}\|\|x_k - x^*\| + \gamma\|S(x^*)\|\|x_k - x^*\| \\
&= O(\|x_k - x^*)\|.
\end{aligned}$$

即

$$\begin{aligned}
\|(J_k^TJ_k)^{-1}S_k\| &\le \|[J(x^*)^TJ(x^*)]^{-1}S(x^*)\| \\
&\quad + \|(J_k^TJ_k)^{-1}S_k - [J(x^*)^TJ(x^*)]^{-1}S(x^*)\| \\
&= \|[J(x^*)^TJ(x^*)]^{-1}S(x^*)\| + O(\|x_k - x^*)\|. \qquad (7.8)
\end{aligned}$$

将 (7.8) 代入 (7.7) 即得本定立的结论. 证毕. □

注 1 若问题 (7.1) 满足定理 33 的条件且最优解 x^* 使得目标函数值取零, 则 $S(x^*) = 0$, 上面的结论表明迭代点列二阶收敛到 x^*. 但当 $F(x)$ 在最优解点的函数值不为 0 时, 由于 $\nabla^2 f(x)$ 略去了不容忽视的项 $S(x)$, 因而难于期待 Gauss-Newton 算法会有好的数值效果.

7.2 Levenberg-Marquardt 方法

Gauss-Newton 算法在迭代过程中要求矩阵 $J(x_k)$ 列满秩, 而这一条件限制了它的应用. 为克服这个困难, Levenberg-Marquardt 方法通过求解下述优化模型来获取搜索方向

$$d_k = \arg\min_{d\in R^n} \quad \|J_kd + F_k\|^2 + \mu_k\|d\|^2,$$

其中 $\mu_k > 0$. 由最优性条件知 d_k 满足

$$\nabla(\|J_kd + F_k\|^2 + \mu_k\|d\|^2) = 2[(J_k^TJ_k + \mu_kI)d + J_k^TF_k] = 0.$$

求得

$$d_k = -(J_k^TJ_k + \mu_kI)^{-1}J_k^TF_k. \qquad (7.9)$$

若 $g_k = J_k^T F_k \neq 0$, 则对任意 $\mu_k > 0$,

$$g_k^T d_k = -(J_k^T F_k)^T (J_k^T J_k + \mu_k I)^{-1} (J_k^T F_k) < 0.$$

所以 d_k 是 $f(x)$ 在 x_k 点的下降方向. 这样, 我们便得到求解非线性最小乘问题的 L-M 方法:

算法 20 (**全局收敛的** L-M **方法**)

步 1 取 ρ, $\sigma \in (0,1)$ 和 $\mu_0 > 0$, 置 $k := 1$.

步 2 若 $g(x_k) = 0$, 停算.

步 3 求解下面的方程组

$$(J_k^T J_k + \mu_k I)d = -J_k^T F_k, \tag{7.10}$$

得 d_k.

步 4 由 Armijio 搜索求步长. 令 m_k 是满足下面不等式的最小非负整数 m :

$$f(x_k + \rho^m d_k) \leq f_k + \sigma \rho^m g_k^T d_k. \tag{7.11}$$

令 $\alpha_k = \rho^{m_k}$.

步 5 置 $x_{k+1} := x_k + \alpha_k d_k$, $k := k+1$, 按某种方式更新 μ_k 的值, 转步 2.

下面我们来讨论 L-M 算法的收敛性. 注意到算法 20 中搜索方向 d_k 的取值其实是与 μ_k 有关的, 严格意义上讲, d_k 应记为 $d_k(\mu_k)$. 因此 L-M 方法的关键是在迭代过程中如何调整参数 μ_k. 为此我们先给出如下结论.

引理 8 $\|d_k(\mu)\|$ 关于 $\mu > 0$ 单调不增, 且当 $\mu \to \infty$ 时, $\|d_k(\mu)\| \to 0$.

证明 注意到

$$\frac{\partial \|d_k(\mu)\|^2}{\partial \mu} = 2 d_k(\mu)^T \frac{\partial d_k(\mu)}{\partial \mu}.$$

由 (7.10) 知

$$(J_k^T J_k + \mu I) d_k(\mu) = -J_k^T F_k.$$

对上式两边关于 μ 求导,

$$d_k(\mu) + (J_k^T J_k + \mu I) \frac{\partial d_k(\mu)}{\partial \mu} = 0.$$

故

$$\frac{\partial d_k(\mu)}{\partial \mu} = -(J_k^T J_k + \mu I)^{-1} d_k(\mu). \tag{7.12}$$

于是,

$$\frac{\partial \|d_k(\mu)\|^2}{\partial \mu} = -2 d_k(\mu)^T (J_k^T J_k + \mu I)^{-1} d_k(\mu) \leq 0. \tag{7.13}$$

从而 $\|d_k(\mu)\|^2$ 关于 μ 单调不增. 由 (7.9) 式可以得到命题的第二个结论. 证毕. □

从几何直观来看, 当矩阵 $J_k^T J_k$ 接近奇异时, 由 Gauss-Newton 算法得到的搜索方向的模 $\|d_k^{GN}\|$ 相当地大. 而在 L-M 方法中, 通过引入正参数 μ 就避免了这种情形出现. 下面讨论参数 μ 对搜索方向角度的影响.

引理 9 $d_k(\mu)$ 与 $-g_k$ 的夹角 θ 关于 $\mu > 0$ 单调不增.

证明 由

$$\cos\theta = \frac{-g_k^T d_k(\mu)}{\|g_k\| \|d_k(\mu)\|}$$

知

$$\begin{aligned}
\frac{\partial \cos\theta}{\partial \mu} &= \frac{\partial}{\partial \mu}\left(\frac{-g_k^T d_k(\mu)}{\|g_k\| \|d_k(\mu)\|}\right) \\
&= \frac{-g_k^T \dfrac{\partial d_k(\mu)}{\partial \mu} \|g_k\| \|d_k(\mu)\| + g_k^T d_k(\mu) \|g_k\| \dfrac{\partial \|d_k(\mu)\|}{\partial \mu}}{\|g_k\|^2 \|d_k(\mu)\|^2}.
\end{aligned} \tag{7.14}$$

利用 (7.12)-(7.13), 将上式中的分子展开

$$\begin{aligned}
& -g_k^T \frac{\partial d_k(\mu)}{\partial \mu} \|g_k\| \|d_k(\mu)\| + g_k^T d_k(\mu) \|g_k\| \frac{\partial \|d_k(\mu)\|}{\partial \mu} \\
&= \|d_k(\mu)\| \|g_k\| g_k^T (J_k^T J_k + \mu I)^{-1} d_k(\mu) \\
&\quad - g_k^T d_k(\mu) \|g_k\| d_k(\mu)^T (J_k^T J_k + \mu I)^{-1} d_k(\mu) / \|d_k(\mu)\| \\
&= \|g_k\| g_k^T \big[\|d_k(\mu)\| (J_k^T J_k + \mu I)^{-1} d_k(\mu) \\
&\quad - d_k(\mu) d_k(\mu)^T (J_k^T J_k + \mu I)^{-1} d_k(\mu) / \|d_k(\mu)\| \big] \\
&= \frac{\|g_k\|}{\|d_k(\mu)\|} g_k^T \big[\|d_k(\mu)\|^2 I - d_k(\mu) d_k(\mu)^T \big] (J_k^T J_k + \mu I)^{-1} d_k(\mu) \\
&= -\frac{\|g_k\|}{\|d_k(\mu)\|} g_k^T \big[\|d_k(\mu)\|^2 I - d_k(\mu) d_k(\mu)^T \big] (J_k^T J_k + \mu I)^{-2} g_k
\end{aligned}$$

$$
\begin{aligned}
&= -\|g_k\|\|d_k(\mu)\|g_k^T(J_k^TJ_k+\mu I)^{-2}g_k \\
&\quad +\|g_k\|g_k^T(J_k^TJ_k+\mu I)^{-1}g_kg_k^T(J_k^TJ_k+\mu I)^{-3}g_k/\|d_k(\mu)\| \\
&= \frac{\|g_k\|}{\|d_k(\mu)\|}\Big\{\big[-g_k^T(J_k^TJ_k+\mu I)^{-2}g_kg_k^T(J_k^TJ_k+\mu I)^{-2}g_k\big] \\
&\quad +g_k^T(J_k^TJ_k+\mu I)^{-1}g_kg_k^T(J_k^TJ_k+\mu I)^{-3}g_k\Big\}.
\end{aligned}
$$

因为 $J_k^TJ_k$ 半正定, 故存在正交阵 Q 使得

$$Q^TJ_k^TJ_kQ = \text{diag}\{\lambda_1,\cdots,\lambda_n\}.$$

记 $v_i := (Q^Tg_k)_i$, 则有

$$Q^T(J_k^TJ_k+\mu I)^{-1}Q = \text{diag}\Big\{\frac{1}{\lambda_1+\mu},\cdots,\frac{1}{\lambda_n+\mu}\Big\},$$

$$
\begin{aligned}
g_k^T(J_k^TJ_k+\mu I)^{-1}g_k &= (Q^Tg_k)^T\text{diag}\Big\{\frac{1}{\lambda_1+\mu},\cdots,\frac{1}{\lambda_n+\mu}\Big\}(Q^Tg_k) \\
&= \sum_{i=1}^n \frac{1}{\lambda_i+\mu}v_i^2.
\end{aligned}
$$

所以

$$
\begin{aligned}
g_k^T(J_k^TJ_k+\mu I)^{-2}g_k &= \sum_{i=1}^n \frac{1}{(\lambda_i+\mu)^2}v_i^2, \\
g_k^T(J_k^TJ_k+\mu I)^{-3}g_k &= \sum_{i=1}^n \frac{1}{(\lambda_i+\mu)^3}v_i^2.
\end{aligned}
$$

这样 (7.14) 式的分子等于

$$
\begin{aligned}
&\frac{\|g_k\|}{\|d_k(\mu)\|}\Big[-\Big(\sum_{i=1}^n\frac{v_i^2}{(\lambda_i+\mu)^2}\Big)^2+\Big(\sum_{i=1}^n\frac{v_i^2}{\lambda_i+\mu}\Big)\Big(\sum_{i=1}^n\frac{v_i^2}{(\lambda_i+\mu)^3}\Big)\Big] \\
&= \frac{\|g_k\|}{\|d_k(\mu)\|}\sum_{i=1}^n\sum_{j=1}^n\Big[\frac{-v_i^2v_j^2}{(\lambda_i+\mu)^2(\lambda_j+\mu)^2}+\frac{v_i^2v_j^2}{(\lambda_i+\mu)(\lambda_j+\mu)^3}\Big] \\
&= \frac{\|g_k\|}{\|d_k(\mu)\|}\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n\frac{v_i^2v_j^2}{(\lambda_i+\mu)^3(\lambda_j+\mu)^3}\big[-2(\lambda_i+\mu)(\lambda_j+\mu) \\
&\quad +(\lambda_i+\mu)^2+(\lambda_j+\mu)^2\big] \;\geq\; 0.
\end{aligned}
$$

从而 d_k 与 $-g_k$ 的夹角 θ 关于 $\mu > 0$ 单调不增. 证毕. □

可以设想, 当参数 $\mu > 0$ 充分大时, $d_k(\mu)$的方向与目标函数的负梯度方向一致.

引理 10 $(J_k^T J_k + \mu I)$ 的条件数关于 $\mu > 0$ 单调不增.

证明 由 $J_k^T J_k$ 为对称半正定矩阵可知, 存在正交阵 Q, 使

$$Q^T J_k^T J_k Q = \mathrm{diag}\{\lambda_1, \cdots, \lambda_n\}, \quad 其中\ \lambda_1 \geq \cdots \geq \lambda_n \geq 0.$$

所以 $(J_k^T J_k + \mu I)$ 的条件数为

$$\|(J_k^T J_k + \mu I)\| \|(J_k^T J_k + \mu I)^{-1}\| = \frac{\lambda_1 + \mu}{\lambda_n + \mu}.$$

而

$$\frac{\partial}{\partial \mu}\Big(\frac{\lambda_1 + \mu}{\lambda_n + \mu}\Big) = \frac{\lambda_n - \lambda_1}{(\lambda_n + \mu)^2} \leq 0,$$

从而 $(J_k^T J_k + \mu I)$ 的条件数关于 $\mu > 0$ 单调不增. 证毕. □

在具体的 L-M 算法中, 我们用类似于调整信赖域半径的策略来调整参数 μ. 首先, 在当前迭代点定义一个二次函数

$$q(d) = f(x_k) + (J_k^T F_k)^T d + \frac{1}{2} d^T (J_k^T J_k) d.$$

基于当前给出的 μ, 根据 (7.9) 计算 d_k, 然后考虑 $q(d)$ 和目标函数的增量

$$\begin{aligned} \Delta q(d_k) &= q(d_k) - q(0) = (J_k^T F_k)^T d_k + \frac{1}{2} d_k^T (J_k^T J_k) d_k, \\ \Delta f(d_k) &= f(x_k + d_k) - f(x_k) = f(x_{k+1}) - f(x_k). \end{aligned}$$

用 η_k 表示两增量之比

$$\eta_k = \frac{\Delta f(d_k)}{\Delta q(d_k)} = \frac{f(x_{k+1}) - f(x_k)}{(J_k^T F_k)^T d_k + \frac{1}{2} d_k^T (J_k^T J_k) d_k}.$$

在 L-M 算法的每一步, 先给 μ_k 一个初始值, 如取为上一次迭代步的值, 计算 d_k. 然后根据 η_k 的值调整 μ_k, 最后根据调整后的 μ_k 计算 d_k, 并进行线搜索, 进而完成 L-M 算法的一个迭代步. 显然, 当 η_k 接近 1 时, 二次函数 $q(d)$ 在 x_k 点拟合目标函数比较好, 用 L-M 方法求解非线性最小乘问题时, 参数 μ 应取得小一些. 换

言之, 此时用 Gauss-Newton 法求解更为有效. 反过来, 当 η_k 接近 0 时, 二次函数 $q(d)$ 在 x_k 点拟合目标函数比较差, 需要减小 d_k 的模长. 根据引理 8, 应增大参数 μ 的取值来限制 d_k 的模长. 而当比值 η_k 既不接近于 0 也不接近于 1, 则认为参数 μ_k 选取得当, 不做调整. 通常 η 的临界值为 0.25 和 0.75. 据此, 得到算法 20 中参数 μ_k 的一个更新规则如下

$$\mu_{k+1} := \begin{cases} 0.1\mu_k, & \text{当 } \eta_k > 0.75, \\ \mu_k, & \text{当 } 0.25 \leq \eta_k \leq 0.75, \\ 10\mu_k, & \text{当 } \eta_k < 0.25. \end{cases} \tag{7.15}$$

下面是 Levenerg-Marquardt 方法的收敛性定理.

定理 34 设 $\{x_k\}$ 是由算法 20 产生无穷迭代序列, 若 $\{x_k, \mu_k\}$ 的某一聚点 (x^*, μ^*) 满足 $J(x^*)^T J(x^*) + \mu^* I$ 正定, 则 $\nabla f(x^*) = J(x^*)^T F(x^*) = 0$.

证明 由于 $\mu_k > 0$, d_k 为下降方向, 存在收敛于 x^* 的子列 x_{k_j}, 满足

$$J_{k_j}^T J_{k_j} \to J(x)^*)^T J(x^*), \quad \mu_{k_j} \to \mu^*.$$

由于 $J(x^*)^T J(x^*) + \mu^* I$ 是正定矩阵, 若 $\nabla f(x^*) \neq 0$, 则

$$d_{k_j} \to d^* = -[J(x^*)^T J(x^*) + \mu^* I]^{-1} J(x^*)^T F(x^*),$$

而且 d^* 是 x^* 点的下降方向. 所以对 $\rho \in (0,1)$, 存在非负整数 m^* 使得

$$f(x^* + \rho^{m^*} d^*) < f(x^*) + \sigma \rho^{m^*} \nabla f(x^*)^T d^*.$$

注意到 $x_{k_j} \to x^*$, 当 j 充分大时, 由连续性知

$$f(x_{k_j} + \rho^{m^*} d_{k_j}) < f(x_{k_j}) + \sigma \rho^{m^*} \nabla f(x_{k_j})^T d_{k_j}.$$

由 Armijo 步长规则知 $m^* \geq m_{k_j}$, 所以

$$\begin{aligned} f(x_{k_j+1}) &= f(x_{k_j} + \rho^{m_{k_j}} d_{k_j}) \\ &\leq f(x_{k_j}) + \sigma \rho^{m_{k_j}} \nabla f(x_{k_j})^T d_{k_j} \\ &\leq f(x_{k_j}) + \sigma \rho^{m^*} \nabla f(x_{k_j})^T d_{k_j}, \end{aligned}$$

即对充分大的 j,

$$f(x_{k_j+1}) \leq f(x_{k_j}) + \sigma \rho^{m^*} \nabla f(x_{k_j})^T d_{k_j}. \tag{7.16}$$

又

$$\lim_{j\to\infty} f(x_{k_j+1}) = \lim_{j\to\infty} f(x_{k_j}) = f(x^*).$$

从而对 (7.16) 两边求极限得

$$f(x^*) \le f(x^*) + \sigma\rho^{m^*}\nabla f(x^*)^T d^*.$$

这与 $\nabla f(x^*)^T d^* < 0$ 矛盾. 所以 $\nabla f(x^*) = 0$. 证毕. □

下面分析算法 20 的收敛速度.

定理 35 设由算法 20 产生的迭代序列 $\{x_k\}$ 收敛到 (7.1) 的一个局部最优解 x^*. 若 $J(x^*)^T J(x^*)$ 非奇异, $\left(\frac{1}{2}-\sigma\right)J(x^*)^T J(x^*) - \frac{1}{2}S(x^*)$ 正定, 且 $G(x) = J(x)^T J(x) + S(x)$ 在 x^* 附近一致连续, $\mu_k \to 0$. 则当 k 充分大时, $\alpha_k = 1$, 且

$$\limsup_{k\to\infty} \frac{\|x_{k+1}-x^*\|}{\|x_k - x^*\|} \le \|[J(x^*)^T J(x^*)]^{-1}\|\|S(x^*)\|. \tag{7.17}$$

证明 要证 $\alpha_k = 1$, 只需证对充分大的 k,

$$f(x_k + d_k) - f(x_k) \le \sigma g_k^T d_k.$$

对任意 $k > 0$, 由中值定理知存在 $\zeta_k \in (0,1)$, 使得

$$f(x_k + d_k) - f(x_k) = g_k^T d_k + \frac{1}{2} d_k^T G(x_k + \zeta_k d_k) d_k.$$

由 (7.9) 式,

$$\begin{aligned}
&-(1-\sigma) g_k^T d_k - \frac{1}{2} d_k^T G(x_k+\zeta_k d_k) d_k \\
&= (1-\sigma) d_k^T (J_k^T J_k + \mu_k I) d_k - \frac{1}{2} d_k^T G(x_k+\zeta_k d_k) d_k \\
&= (1-\sigma) d_k^T J_k^T J_k d_k + (1-\sigma)\mu_k \|d_k\|^2 - \frac{1}{2} d_k^T (J_k^T J_k + S_k) d_k \\
&\quad + \frac{1}{2} d_k^T [G(x_k) - G(x_k+\zeta_k d_k)] d_k \\
&= \left(\frac{1}{2}-\sigma\right) d_k^T J_k^T J_k d_k + (1-\sigma)\mu_k \|d_k\|^2 - \frac{1}{2} d_k^T S_k d_k \\
&\quad + \frac{1}{2} d_k^T [G(x_k) - G(x_k+\zeta_k d_k)] d_k.
\end{aligned}$$

由 $x_k \to x^*$ 知 $g_k \to 0$, 从而 $d_k \to 0$. 又 $\mu_k \to 0$, 所以利用 $G(x)$ 的一致连续性知

$$V_k := (1-\sigma)\mu_k I + \frac{1}{2}[G(x_k) - G(x_k+\zeta_k d_k)] \to 0.$$

由题设, k 充分大时,

$$
\begin{aligned}
&-(1-\sigma)g_k^T d_k - \frac{1}{2}d_k^T G(x_k+\zeta_k d_k)d_k \\
&= d_k^T\Big[\Big(\frac{1}{2}-\sigma\Big)J_k^T J_k - \frac{1}{2}S_k\Big]d_k + d_k^T V_k d_k > 0.
\end{aligned}
$$

从而 $\alpha_k = 1$ 成立. 这样,

$$
\begin{aligned}
x_{k+1}-x^* &= x_k - x^* - (J_k^T J_k+\mu_k I)^{-1} g_k \\
&= -(J_k^T J_k+\mu_k I)^{-1}\{[g_k - G_k(x_k-x^*)] \\
&\quad -[(J_k^T J_k+\mu_k I)(x_k-x^*) - G_k(x_k-x^*)]\} \\
&= -(J_k^T J_k+\mu_k I)^{-1}[g_k - g(x^*) - G_k(x_k-x^*)] \\
&\quad +(J_k^T J_k+\mu_k I)^{-1}(\mu_k I - S_k)(x_k-x^*) \\
&= -(J_k^T J_k+\mu_k I)^{-1}\int_0^1 \big[G(x^*+t(x_k-x^*)-G(x_k)\big](x_k-x^*)\mathrm{d}t \\
&\quad +(J_k^T J_k+\mu_k I)^{-1}(\mu_k I - S_k)(x_k-x^*).
\end{aligned}
$$

故

$$
\begin{aligned}
\|x_{k+1}-x^*\| \ \le\ & \|(J_k^T J_k+\mu_k I)^{-1}\|\Big[\int_0^1 \big\|G(x^*+t(x_k-x^*))-G(x_k)\big\|\mathrm{d}t \\
& +\|\mu_k I - S_k\|\Big]\|x_k-x^*\|..
\end{aligned}
$$

由 $\mu_k \to 0$, $x_k \to x^*$ 和 $G(x)$ 的一致连续性,

$$
\limsup_{k\to\infty}\frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|} \le \|[J(x^*)^T J(x^*)]^{-1}\|\|S(x^*)\|.
$$

证毕. □

注 2　若 (7.1) 的目标函数的最优值为 0 时, 若取

$$
\mu_k = \|F(x_k)\|^{1+\sigma}, \quad \sigma\in[0,1],
$$

或

$$
\mu_k = \theta\|F_k\| + (1-\theta)\|J_k^T F_k\|, \quad \theta\in[0,1],
$$

则可以建立 Levenberg-Marquardt 算法的二阶收敛性质.

7.3　L-M 算法的 Matlab 程序

本节我们给出 L-M 算法 20 的 Matlab 程序, 在某种意义上该程序是通用的.

程序 14　利用 LM 方法求解非线性方程组 $F(x)=0$, 可适用于未知数的个数与方程的个数不相等的情形.

```
function [x,val,k]=lmm(Fk,JFk,x0)
%功能: 用L-M方法求解非线性方程组: F(x)=0
%输入: x0是初始点, Fk, JFk 分别是求F(xk)及F'(xk)的函数.
%输出: x, val分别是近似解及||F(xk)||的值, k是迭代次数.
maxk=100;   %给出最大迭代次数
rho=0.55;sigma=0.4; muk=norm(feval(Fk,x0));
k=0;  epsilon=1e-6; n=length(x0);
while(k<maxk)
    fk=feval(Fk,x0); %计算函数值
    jfk=feval(JFk,x0); %计算Jacobi阵
    gk=jfk'*fk;
    dk=-(jfk'*jfk+muk*eye(n))\gk; %解方程组Gk*dk=-gk, 计算搜索方向
    if(norm(gk)<epsilon), break; end  %检验终止准则
    m=0; mk=0;
    while(m<20)   % 用Armijo搜索求步长
        newf=0.5*norm(feval(Fk,x0+rho^m*dk))^2;
        oldf=0.5*norm(feval(Fk,x0))^2;
        if(newf<oldf+sigma*rho^m*gk'*dk)
            mk=m; break;
        end
        m=m+1;
    end
    x0=x0+rho^mk*dk;
    muk=norm(feval(Fk,x0));
    k=k+1;
end
x=x0;
val=0.5*muk^2;
```

我们利用上面的程序来求解一个信非线性方程组.

例 21　用 LM 方法求解非线性方程组求解下列方程组

$$\begin{cases} x_1 - 0.7\sin x_1 - 0.2\cos x_2 = 0, \\ x_2 - 0.7\cos x_1 + 0.2\sin x_2 = 0. \end{cases}$$

解　先编制两个函数程序, 并将它们作为 m 文件与程序 14 一起存放于当前工作目录下.

```
%Fk.m
function y=Fk(x)
y(1)=x(1)-0.7*sin(x(1))-0.2*cos(x(2));
y(2)=x(2)-0.7*cos(x(1))+0.2*sin(x(2));
y=y(:);
%JFk.m
function JF=JFk(x)
JF=[1-0.7*cos(x(1)), 0.2*sin(x(2));
   0.7*sin(x(1)), 1+0.2*cos(x(2))];
```

该问题有解 $x^* \approx (0.52652, 0.50792)^T$. 我们利用程序 14, 终止准则取为 $\|\nabla f(x_k)\| \le 10^{-6}$. 取不同的初始点, 数值结果如下表.

表 8.1 L-M 方法的数值结果.

初始点 (x_0)	迭代次数 (k)	目标函数值 $\|F(x_k)\|^2/2$
$(0.0, 0.0)^T$	7	9.4380×10^{-16}
$(1.0, 1.0)^T$	6	7.4433×10^{-19}
$(1.0, -1.0)^T$	9	4.6783×10^{-19}
$(-1.0, 1.0)^T$	10	7.6358×10^{-22}
$(5.0, 5.0)^T$	14	3.2383×10^{-20}
$(-5.0, -5.0)^T$	20	2.1319×10^{-19}

习　题　7

1. 设有非线性方程组

$$f_1(x) = x_1^3 - 2x_2^2 - 1 = 0,$$
$$f_2(x) = 2x_1 + x_2 - 2.$$

(1) 列出求解这个方程组的非线性最小二乘问题的数学模型.

(2) 写出求解该问题的高斯-牛顿法迭代公式的具体形式.

(3) 初始点取为 $x_0=(2,2)^T$, 迭代三次.

2. 已知某物理量 y 与另外两个物理量 t_1,t_2 的关系为

$$y=\frac{x_1x_3t_1}{1+x_1t_1+x_2t_2},$$

其中 x_1,x_2,x_3 是待定参数. 为确定这三个参数，测得 t_1,t_2 和 y 的一组数据.

t_1	1.0	2.0	1.0	2.0	0.1
t_2	1.0	1.0	2.0	2.0	0.0
y	0.13	0.22	0.08	0.13	0.19

(1) 用最小二乘法建立关于确定 x_1,x_2,x_3 的数学模型.

(2) 对列出的非线性最小二乘问题写出高斯-牛顿法迭代公式的具体形式.

3. 设 $r_1(x)=x_2-x_1^2$, $r_2(x)=1-x_2$, $r(x)=(r_1(x),r_2(x))^T$,

$$f(x)=\frac{1}{2}[r_1(x)^2+r_2(x)^2]=\frac{1}{2}r(x)^Tr(x),$$

已知该问题 $\min f(x)$ 的极小点为 $x^*=(1,1)^T$.

(1) 对于任何初始点 x_0, 写出按如下牛顿迭代公式求极小点的计算步骤:

$$\begin{cases} x_{k+1}=x_k+\alpha_kd_k \\ d_k=-[\nabla^2f(x_k)]^{-1}\nabla f(x_k), & k=0,1,\cdots \\ \alpha_k:f(x_k+\alpha_kd_k)=\min f(x_k+\alpha d_k) \end{cases}$$

(2) 试证明: 当 $x\to x^*$ 时, $\nabla^2f(x)\to\nabla r(x)\nabla r(x)^T$.

4. 考虑非线性方程组:

$$\begin{cases} r_1(x)=x_1^3-x_2-1=0, \\ r_2(x)=x_1^2-x_2=0 \end{cases}$$

及平方和函数

$$f(x)=\frac{1}{2}[r_1(x)^2+r_2(x)^2].$$

证明: $x=(1.46557,2.14790)^T$ 是 $f(x)$ 的全局极小点, 并且是非线性方程组的解; $x=(0,-0.5)^T$ 是 $f(x)$ 的局部极小点; $x=(2/3,-7/54)^T$ 是 $f(x)$ 的鞍点.

5. 设 x_1,x_2 分别是方程组 $(A^TA+\mu_iI)x=-A^Tr$, $i=1,2$, 对应于 μ_1,μ_2 的解, 其中 $\mu_1>\mu_2>0$, $A\in R^{m\times n}$, $r\in R^m$. 试证明 $\|Ax_2+r\|_2^2<\|Ax_1+r\|_2^2$.

6. 利用 L-M 方法的 Matlab 程序求解

$$\min f(x) = \frac{1}{2}\sum_{i=1}^{5} r_i(x)^2,$$

其中

$$\begin{aligned} &r_1(x) = x_1^2 + x_2^2 + x_3^2 - 1, \quad r_2(x) = x_1 + x_2 + x_3 - 1, \\ &r_3(x) = x_1^2 + x_2^2 + (x_3 - 2)^2 - 1, \quad r_4(x) = x_1 + x_2 - x_3 + 1, \\ &r_5(x) = x_1^3 + 3x_2^2 + (5x_3 - x_1 + 1)^2 - 36t, \end{aligned}$$

t 是参数, 可取 $t = 0.5$, 1, 5 等. 注意 $t = 1$时, $x^* = (0,0,1)^T$ 是全局极小点, 这时问题为零残量, 比较不同参数的计算效果.

第八章　最优性条件

8.1　等式约束问题的最优性条件

本节讨论的最优性条件适合于下面的等式约束问题

$$\begin{aligned} &\min\ f(x), \\ &\text{s.t.}\ \ h_i(x)=0,\ i=1,2,\cdots,l. \end{aligned} \tag{8.1}$$

为了研究问题的方便, 我们作问题 (8.1) 的所谓拉格朗日函数

$$L(x,\lambda)=f(x)-\sum_{i=1}^{l}\lambda_i h_i(x), \tag{8.2}$$

其中 $\lambda=(\lambda_1,\lambda_2,\cdots,\lambda_l)^T$ 称为乘子向量.

下面的拉格朗日定理描述了问题 (8.1) 取极小值的一阶必要条件, 也就是所谓的 KT 条件 (Kuhn-Tucker 条件).

定理 36 （**拉格朗日定理**） 假设 x^* 是问题 (8.1) 的局部极小点, $f(x)$ 和 $h_i(x)\,(i=1,2,\cdots,l)$ 在 x^* 的某邻域内连续可微. 若向量组 $\nabla h_i(x^*)\,(i=1,2,\cdots,l)$ 线性无关, 则存在乘子向量 $\lambda^*=(\lambda_1^*,\lambda_2^*,\cdots,\lambda_l^*)^T$ 使得

$$\nabla_x L(x^*,\lambda^*)=0,$$

即

$$\nabla f(x^*)-\sum_{i=1}^{l}\lambda_i^*\nabla h_i(x^*)=0.$$

证 记

$$H=\big(\nabla h_1(x^*),\nabla h_2(x^*),\cdots,\nabla h_l(x^*)\big).$$

由定理的假设知 H 列满秩. 因此, 若 $l=n$, 则 H 是可逆方阵, 从而矩阵 H 的列构成 $\mathbb{R}^n$ 中的一组基, 故存在 $\lambda^*\in\mathbb{R}^l\,(l=n)$ 使得

$$\nabla f(x^*)=\sum_{i=1}^{l}\lambda_i^*\nabla h_i(x^*),$$

此时结论得证.

下面设 $l<n$. 不失一般性, 可设 H 的前 l 行构成的 l 阶子矩阵 H_1 是非奇异的. 据此, 将 H 分块为

$$H=\begin{pmatrix} H_1\\ H_2\end{pmatrix}.$$

令 $y=(x_1,\cdots,x_l)^T$, $z=(x_{l+1},\cdots,x_n)^T$, 并记 $h(x)=(h_1(x),\cdots,h_l(x))^T$. 则有 $h(y^*,z^*)=0$, 且 $h(y,z)$ 在点 (y^*,z^*) 关于 y 的 Jacobi 矩阵 $H_1^T=\nabla_y h(y^*,z^*)$ 可逆. 故由隐函数定理可知, 在 z^* 附近存在关于 z 的连续可微函数 $y=u(z)$ 使得

$$h(u(z),z)=0.$$

对上式两边关于 z 求导得

$$\nabla_y h(u(z),z)\nabla u(z)+\nabla_z h(u(z),z)=0,$$

故

$$\nabla u(z^*)=-H_1^{-T}H_2^T. \tag{8.3}$$

在 z^* 附近, 由$h(u(z),z)=0$ 知 z^* 是无约束优化问题

$$\min_{z\in R^{n-l}} f(u(z),z)$$

的局部极小点, 故有

$$\nabla_z f(u(z^*),z^*)=0,$$

即

$$\nabla u(z^*)^T\nabla_y f(y^*,z^*)+\nabla_z f(y^*,z^*)=0.$$

注意到 $x^*=(y^*,z^*)$, 将 (8.3) 代入上式得

$$-H_2H_1^{-1}\nabla_y f(x^*)+\nabla_z f(x^*)=0.$$

令 $\lambda^* = H_1^{-1}\nabla_y f(x^*)$, 则有

$$\nabla_y f(x^*) = H_1\lambda^*, \quad \nabla_z f(x^*) = H_2\lambda^*.$$

两式合起来即

$$\nabla f(x^*) = \begin{pmatrix} \nabla_y f(x^*) \\ \nabla_z f(x^*) \end{pmatrix} = \begin{pmatrix} H_1 \\ H_2 \end{pmatrix} \lambda^* = \sum_{i=1}^{l} \lambda_i \nabla h_i(x^*).$$

至此已经证明了定理的结论. □

为了讨论等式约束问题的二阶必要条件, 需要用到 (8.2) 定义的拉格朗日函数 $L(x,\lambda)$ 的梯度和关于 x 的 Hesse 阵. 下面, 我们计算出它们的表达式如下:

$$\nabla L(x,\lambda) = \begin{pmatrix} \nabla_x L(x,\lambda) \\ \nabla_\lambda L(x,\lambda) \end{pmatrix} = \begin{pmatrix} \nabla f(x) - \sum\limits_{i=1}^{l} \lambda_i \nabla h_i(x) \\ -h(x) \end{pmatrix},$$

$$\nabla_{xx}^2 L(x,\lambda) = \nabla^2 f(x) - \sum_{i=1}^{l} \lambda_i \nabla^2 h_i(x).$$

如果目标函数和约束函数都是二阶连续可微的, 则可考虑二阶充分性条件.

定理 37 对于等式约束问题 (8.1), 假设 $f(x)$ 和 $h_i(x)\,(i=1,2,\cdots,l)$ 都是二阶连续可微的, 并且存在 $(x^*,\lambda^*) \in \mathbb{R}^n \times \mathbb{R}^l$ 使得 $\nabla L(x^*,\lambda^*) = 0$. 若对任意的 $0 \neq d \in \mathbb{R}^n$, $\nabla h_i(x^*)^T d = 0\,(i=1,2,\cdots,l)$, 均有 $d^T \nabla_{xx}^2 L(x^*,\lambda^*) d > 0$, 则 x^* 是问题 (8.1) 的一个严格局部极小点.

证 用反证法. 若 x^* 不是严格局部极小点, 则必存在邻域 $N(x^*,\delta)$ 及收敛于 x^* 的序列 $\{x_k\}$, 使得 $x_k \in N(x^*,\delta)$, $x_k \neq x^*$, 且有

$$f(x^*) \geq f(x_k), \;\; h_i(x_k) = 0, \; i = 1,2,\cdots,l, \; k = 1,2,\cdots$$

令 $x_k = x^* + \alpha_k z_k$, 其中 $\alpha_k > 0$, $\|z_k\| = 1$, 序列 $\{(\alpha_k, z_k)\}$ 有子列收敛于 $(0, z^*)$ 且 $\|z^*\| = 1$.

由泰勒中值公式得

$$0 = h_i(x_k) - h_i(x^*) = \alpha_k z_k^T \nabla h_i(x^* + \theta_{ik}\alpha_k z_k),$$

其中 $\theta_{ik} \in (0,1)$. 上式两边同除以 α_k, 并令 $k \to \infty$ 得

$$\nabla h_i(x^*)^T z^* = 0,\ \ i = 1, 2, \cdots, l. \tag{8.4}$$

再由泰勒展开式得

$$L(x_k, \lambda^*) = L(x^*, \lambda^*) + \alpha_k \nabla_x L(x^*, \lambda^*)^T z_k + \frac{1}{2}\alpha_k^2 z_k^T \nabla_{xx}^2 L(x^*, \lambda^*) z_k + o(\alpha_k^2).$$

由于 x_k 都满足等式约束, 故有

$$\begin{aligned} 0 &\geq f(x_k) - f(x^*) = L(x_k, \lambda^*) - L(x^*, \lambda^*) \\ &= \frac{1}{2}\alpha_k^2 z_k^T \nabla_{xx}^2 L(x^*, \lambda^*) z_k + o(\alpha_k^2). \end{aligned}$$

上式两边同除以 $\alpha_k/2$, 可得

$$z_k^T \nabla_{xx}^2 L(x^*, \lambda^*) z_k + \frac{o(2\alpha_k^2)}{\alpha_k^2} \leq 0.$$

对上式取极限 $(k \to \infty)$ 即得

$$(z^*)^T \nabla_{xx}^2 L(x^*, \lambda^*) z^* \leq 0.$$

由于 z^* 满足 (8.4), 故得出矛盾. 因此 x^* 一定严格局部极小点. □

8.2　不等式约束问题的最优性条件

本小节我们考虑不等式约束优化问题的最优性条件:

$$\begin{aligned} &\min\ \ f(x), \\ &\text{s.t.}\ \ g_i(x) \geq 0,\ i = 1, 2, \cdots, m. \end{aligned} \tag{8.5}$$

记可行域为 $\mathcal{D} = \{x \in \mathbb{R}^n |\, g_i(x) \geq 0, i = 1, 2, \cdots, n\}$, 指标集 $I = \{1, \cdots, m\}$.

不等式约束问题的最优性条件需要用到所谓的有效约束和非有效约束的概念. 对于一个可行点 $\bar{x}$, 即 $\bar{x} \in \mathcal{D}$. 此时可能会出现两种情形. 即有些约束函数满足 $g_i(\bar{x}) = 0$, 而另一些约束函数则满足 $g_i(\bar{x}) > 0$. 对于后一种情形, 在 $\bar{x}$ 的某个邻域内仍然保持 $g_i(\bar{x}) > 0$ 成立, 而前者则不具备这种性质. 因此有必要把这两种情形区分开来.

定义 17 若问题 (8.5) 的一个可行点 $\bar{x} \in \mathcal{D}$ 使得 $g_i(\bar{x}) = 0$, 则称不等式约束 $g_i(x) \geq 0$ 为 $\bar{x}$ 的有效约束. 反之, 若有 $g_i(\bar{x}) > 0$, 则称不等式约束 $g_i(x) \geq 0$ 为 $\bar{x}$ 的非有效约束. 称集合

$$I(\bar{x}) = \{i : g_i(\bar{x}) = 0\} \tag{8.6}$$

为 $\bar{x}$ 处的有效约束指标集, 简称 x 处的有效集 (或积极集).

下面的两个引理是研究不等式约束问题最优性条件的基础.

引理 11 (**Farkas 引理**) 设 $a, b_i \in \mathbb{R}^n \, (i = 1, \cdots, r)$. 则线性不等式组

$$b_i^T d \geq 0, \;\; i = 1, \cdots, r, \; d \in \mathbb{R}^n$$

与不等式

$$a^T d \geq 0$$

相容的充要条件是存在非负实数 $\alpha_1, \cdots, \alpha_r$, 使得 $a = \sum\limits_{i=1}^{r} \alpha_i b_i$.

证 充分性. 即存在非负实数 $\alpha_1, \cdots, \alpha_r$, 使得 $a = \sum\limits_{i=1}^{r} \alpha_i b_i$. 设 $d \in \mathbb{R}^n$ 满足 $b_i^T d \geq 0 \, (i = 1, \cdots, r)$. 那么有

$$a^T d = \sum_{i=1}^{r} \alpha_i b_i^T d \geq 0.$$

必要性. 设所有满足 $b_i^T d \geq 0 \, (i = 1, \cdots, r)$ 的向量 d 同时也满足 $a^T d \geq 0$. 用反证法. 设结论不成立, 即

$$a \notin C = \Big\{ x \in \mathbb{R}^n \Big| \, x = \sum_{i=1}^{r} \alpha_i b_i, \, \alpha_i \geq 0, \, i = 1, \cdots, r \Big\}.$$

设 $a_0 \in C$ 是向量 a 在凸锥 C 上的投影, 即

$$\|a_0 - a\|_2 = \min_{x \in C} \|x - a\|_2,$$

则有 $a_0^T (a_0 - a) = 0$.

(1) 先证明对于任意的 $p \in C$ 必有 $u^T (a_0 - a) \geq 0$. 事实上, 若不然, 则存在一个 $u \in C$, 使 $u^T (a_0 - a) < 0$. 令 $\bar{u} = \dfrac{u}{\|u\|}$, 则 $\bar{u}^T (a_0 - a) = -\tau, \tau > 0$. 注意到 C 是凸锥且 $a_0, \bar{u} \in C$, 故 $a_0 + \tau \bar{u} \in C$. 此时有

$$\|a_0 + \tau \bar{u} - a\|^2 - \|a_0 - a\|^2 = -\tau^2 < 0,$$

这与 a_0 是投影的假设矛盾, 故必有 $u^T(a_0-a)\geq 0,\ \forall\, u\in C$.

(2) 现取 $d=a_0-a$. 由于 $b_i\in C$, 那么由 (1) 的结论可得 $b_i^Td\geq 0$. 故由必要性的假设应有 $a^Td\geq 0$. 但另一方面, 有

$$\begin{aligned} a^Td &= a^T(a_0-a)=a^T(a_0-a)-a_0^T(a_0-a) \\ &= -(a_0-a)^T(a_0-a)=-\|a_0-a\|^2<0, \end{aligned}$$

这与假设矛盾, 必要性得证. □

下面的 Gordan 引理可以认为是 Farkas 引理的一个推论.

引理 12 （**Gordan 引理**） 设 $b_i\in\mathbb{R}^n\,(i=1,\cdots,r)$. 线性不等式组

$$b_i^Td<0,\ \ i=1,2,\cdots,r,\ d\in\mathbb{R}^n \tag{8.7}$$

无解的充要条件是 $b_i\,(i=1,\cdots,r)$ 线性相关, 即存在不全为 0 的非负实数 $\alpha_i\,(i=1,\cdots,r)$, 使得

$$\sum_{i=1}^{r}\alpha_ib_i=0. \tag{8.8}$$

证 充分性. 用反证法. 设 (8.7) 有解, 即存在某个 d_0, 使得 $b_i^Td_0<0,\ \ i=1,\cdots,r$. 于是对于任意不全为 0 的非负实数 $\alpha_i\,(i=1,\cdots,r)$, 有 $\sum_{i=1}^{r}\alpha_ib_i<0$. 另一方面, 由充分性条件 (8.8) 有 $\sum_{i=1}^{r}\alpha_ib_i=0$, 矛盾, 故充分性得证.

必要性. 设不等式组 (8.7) 无解. 故对于任意的 $d\in\mathbb{R}^n$, 至少存在一个指标 i 满足 $b_i^Td\geq 0$. 记 $\beta_0=\max\limits_{1\leq i\leq r}\{b_i^Td\}$, 则必有 $\beta_0\geq 0$ 且

$$\beta_0-b_i^Td\geq 0,\ \ i=1,\cdots,r.$$

下面我们构造 $r+2$ 个 $n+1$ 维向量:

$$\bar{d}=\begin{pmatrix}\beta\\ d\end{pmatrix},\quad \bar{a}=\begin{pmatrix}1\\ 0\end{pmatrix},\quad \bar{b}_i=\begin{pmatrix}1\\ -b_i\end{pmatrix},\quad i=1,\cdots,r, \tag{8.9}$$

其中 $\beta\geq\beta_0$. 那么不难验证上述向量满足 Farkas 引理的条件, 即

$$\bar{b}_i^T\bar{d}=\beta-b_i^Td\geq\beta_0-b_i^Td\geq 0,\ \ i=1,\cdots,r,$$

且 $\bar{a}^T\bar{d}=\beta\geq 0$. 故由 Farkas 引理, 存在非负实数 $\alpha_1,\cdots,\alpha_r$, 使得

$$\bar{a}=\sum_{i=1}^{r}\alpha_i\bar{b}_i.$$

将 (8.9) 代入上式即得

$$\sum_{i=1}^{r}\alpha_i b_i=0,\quad \sum_{i=1}^{r}\alpha_i=1.$$

必要性得证. □

下面的引理可认为是一个几何最优性条件.

引理 13 设 x^* 是不等式约束问题 (8.5) 的一个局部极小点, $I(x^*)=\{i|\,g_i(x^*)=0,\ i=1,\cdots,m\}$. 假设 $f(x)$ 和 $g_i(x)\,(i\in I(x^*))$ 在 x^* 处可微, 且 $g_i(x)\,(i\in I\backslash I(x^*))$ 在 x^* 处连续. 则问题 (8.5) 的可行方向集 $\mathcal{F}$ 与下降方向集 $\mathcal{S}$ 的交集是空集, 即 $\mathcal{F}\cap\mathcal{S}=\emptyset$, 其中

$$\mathcal{F}=\{d\in\mathbb{R}^n|\,\nabla g_i(x^*)^Td>0,\ i\in I(x^*)\},\quad \mathcal{S}=\{d\in\mathbb{R}^n|\,\nabla f(x^*)^Td<0\}.\tag{8.10}$$

证 用反证法. 设 $\mathcal{F}\cap\mathcal{S}\neq\emptyset$, 则存在 $d\in\mathcal{F}\cap\mathcal{S}=\emptyset$. 显然 $d\neq 0$. 由 $\mathcal{F},\mathcal{S}$ 的定义及函数的连续性知, 存在充分小的正数 $\bar{\varepsilon}$, 使得对任意的 $0<\varepsilon\ll\bar{\varepsilon}$, 有

$$f(x^*+\varepsilon d)<f(x^*),\quad g_i(x^*+\varepsilon d)\geq 0,\ \ i=1,\cdots,m.$$

这与假设矛盾. □

下面我们给出不等式约束问题 (8.5) 的一阶必要条件, 即著名的 KT 条件.

定理 38（KT 条件） 设 x^* 是不等式约束问题 (8.5) 的局部极小点, 有效约束集 $I(x^*)=\{i|\,g_i(x^*)=0,i=1,\cdots,m\}$. 并设 $f(x)$ 和 $g_i(x)\,(i=1,\cdots,m)$ 在 x^* 处可微. 若向量组 $\nabla g_i(x^*)\,(i\in I(x^*))$ 线性无关, 则存在向量 $\lambda^*=(\lambda_1^*,\cdots,\lambda_m^*)^T$ 使得

$$\begin{cases}\nabla f(x^*)-\displaystyle\sum_{i=1}^{m}\lambda_i^*\nabla g_i(x^*)=0,\\ g_i(x^*)\geq 0,\ \ \lambda_i^*\geq 0,\ \ \lambda_i^*g_i(x^*)=0,\ i=1,\cdots,m.\end{cases}$$

证 因 x^* 是问题 (8.5) 的局部极小点, 故由引理 13 知, 不存在 $d\in\mathbb{R}^n$ 使得

$$\nabla f(x^*)^Td<0,\quad \nabla g_i(x^*)^Td>0,\ i\in I(x^*),$$

即线性不等式组

$$\nabla f(x^*)^T d < 0, \quad -\nabla g_i(x^*)^T d < 0,\ i \in I(x^*)$$

无解. 于是由 Gordan 引理知, 存在不全为 0 的非负实数 $\mu_0 \geq 0$ 及 $\mu_i \geq 0\,(i \in I(x^*))$, 使得

$$\mu_0 \nabla f(x^*) - \sum_{i \in I(x^*)} \mu_i \nabla g_i(x^*) = 0.$$

不难证明 $\mu_0 \neq 0$. 事实上, 若 $\mu_0 = 0$, 则有 $\sum\limits_{i \in I(x^*)} \mu_i \nabla g_i(x^*) = 0$, 由此可知 $\nabla g_i(x^*)\,(i \in I(x^*))$ 线性相关, 这与假设矛盾. 因此必有 $\mu_0 > 0$. 于是可令

$$\lambda_i^* = \frac{\mu_i}{\mu_0},\ \ i \in I(x^*);\ \ \lambda_i^* = 0,\ \ i \in I \backslash I(x^*),$$

则得

$$\nabla f(x^*) - \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) = 0,$$

及

$$g_i(x^*) \geq 0,\ \ \lambda_i^* \geq 0,\ \ \lambda_i^* g_i(x^*) = 0,\ i = 1, \cdots, m.$$

定理得证. □

8.3 一般约束问题的最优性条件

我们现在考虑一般约束优化问题的最优性条件:

$$\begin{aligned} &\min\ f(x), \\ &\text{s.t.}\ \ h_i(x) = 0,\ i = 1, 2, \cdots, l, \\ &\qquad g_i(x) \geq 0,\ i = 1, 2, \cdots, m. \end{aligned} \tag{8.11}$$

记指标集 $E = \{1, \cdots, l\}$, $I = \{1, \cdots, m\}$. 可行域为 $\mathcal{D} = \{x \in \mathbb{R}^n |\, h_i(x) = 0,\ i \in E,\ g_i(x) \geq 0,\ i \in I\}$.

把定理 36 和定理 38 结合起来即得到一般约束问题 (8.11) 的 KT 一阶必要条件.

定理 39（**KT 一阶必要条件**）设 x^* 是一般约束问题 (8.11) 的局部极小点，在 x^* 处的有效约束集为

$$S(x^*) = E \cup I(x^*) = E \cup \{i \,|\, g_i(x^*) = 0, i \in I\}. \tag{8.12}$$

并设 $f(x)$，$h_i(x)\,(i \in E)$ 和 $g_i(x)\,(i \in I)$ 在 x^* 处可微. 若向量组 $\nabla h_i(x^*)\,(i \in E)$，$\nabla g_i(x^*)\,(i \in I(x^*))$ 线性无关，则存在向量 $(\mu^*, \lambda^*) \in \mathbb{R}^l \times \mathbb{R}^m$，其中 $\mu^* = (\mu_1^*, \cdots, \mu_l^*)^T$，$\lambda^* = (\lambda_1^*, \cdots, \lambda_m^*)^T$，使得

$$\begin{cases} \nabla f(x^*) - \sum\limits_{i=1}^{l} \mu_i^* \nabla h_i(x^*) - \sum\limits_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) = 0, \\ h_i(x^*) = 0,\ i \in E, \\ g_i(x^*) \geq 0,\ \ \lambda_i^* \geq 0,\ \ \lambda_i^* g_i(x^*) = 0,\ i \in I. \end{cases} \tag{8.13}$$

注 3 (1) 称 (8.13) 为 KT 条件，满足这一条件的点 x^* 称为 KT 点. 而把 $(x^*, (\mu^*, \lambda^*))$ 称为 KT 对，其中 (μ^*, λ^*) 称为问题的拉格朗日乘子. 通常 KT 点、KT 对和 KT 条件可以不加区别的使用.

(2) 称 $\lambda_i^* g_i(x^*) = 0\,(i \in I(x^*))$ 为互补性松弛条件. 这意味着 λ_i^* 和 $g_i(x^*)$ 中至少有一个必为 0. 若二者中的一个为 0，而另一个严格大于 0，则称之为满足严格互补性松弛条件.

与等式约束问题相仿，可以定义问题 (8.11) 的拉格朗日函数

$$L(x, \lambda, \mu) = f(x) - \sum_{i=1}^{l} \mu_i h_i(x) - \sum_{i=1}^{m} \lambda_i g_i(x). \tag{8.14}$$

不难求出它关于变量 x 的梯度和 Hesse 阵分别为

$$\nabla_x L(x, \lambda, \mu) = \nabla f(x) - \sum_{i=1}^{l} \mu_i \nabla h_i(x) - \sum_{i=1}^{m} \lambda_i \nabla g_i(x),$$

$$\nabla_{xx}^2 L(x, \lambda, \mu) = \nabla^2 f(x) - \sum_{i=1}^{l} \mu_i \nabla^2 h_i(x) - \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(x).$$

与定理 37 的证明相类似，证明问题 (8.13) 的二阶充分条件:

定理 40 对于约束优化问题 (8.11), 假设 $f(x)$, $g_i(x)\,(i\in I)$ 和 $h_i(x)\,(i\in E)$ 都是二阶连续可微的, 有效约束集 $I(x^*)$ 由 (8.12) 所定义. 且 $(x^*,(\mu^*,\lambda^*))$ 是问题 (8.11) 的 KT 点. 若对任意的 $0\neq d\in\mathbb{R}^n$, $\nabla g_i(x^*)^T d=0\,(i\in I(x^*)$, $\nabla h_i(x^*)^T d=0\,(i\in E)$, 均有 $d^T\nabla^2_{xx}L(x^*,\lambda^*)d>0$, 则 x^* 是问题 (8.11) 的一个严格局部极小点.

例 22 考虑优化问题

$$\begin{aligned}
&\min\ f(x)=-2x_1^2-x_2^2,\\
&\text{s.t.}\ x_1^2+x_2^2-2=0,\\
&\qquad -x_1+x_2\ge 0,\\
&\qquad x_1\ge 0,\ x_2\ge 0.
\end{aligned}$$

试验证 $x^*=(1,1)^T$ 为 KT 点, 并求出问题的 KT 对.

解 计算

$$\nabla f(x^*)=\begin{pmatrix}-4x_1\\-2x_2\end{pmatrix}\Bigg|_{x=x^*}=\begin{pmatrix}-4\\-2\end{pmatrix},\quad \nabla h(x^*)=\begin{pmatrix}2\\2\end{pmatrix},\quad \nabla g_1(x^*)=\begin{pmatrix}-1\\1\end{pmatrix}.$$

令

$$\nabla f(x^*)-\mu^*\nabla h(x^*)-\lambda_1^*\nabla g_1(x^*)=0,$$

解得 $\mu^*=-1.5$, $\lambda_1^*=1$. 再令 $\lambda_2^*=\lambda_3^*=0$, 得

$$\begin{cases}\nabla f(x^*)-\mu^*\nabla h(x^*)-\sum\limits_{i=1}^{3}\lambda_i^*\nabla g_i(x^*)=0,\\ \lambda_i^*g_i(x^*)=0,\ \lambda_i\ge 0,\ i=1,2,3.\end{cases}$$

这表明 x^* 是 KT 点, $(x^*,(\mu^*,\lambda^*))$ 是 KT 对, 其中 $\mu^*=-1.5$, $\lambda^*=(1,0,0)^T$.

一般而言, 问题 (8.11) 的 KT 点不一定是局部极小点. 但如果问题是下面的所谓凸优化问题, 则 KT 点、局部极小点、全局极小点三者是等价的.

下面首先给出约束凸优化问题的定义.

定义 18 对于约束最优化问题

$$\begin{aligned}
&\min\ f(x),\ \ x\in\mathbb{R}^n\\
&\text{s.t.}\begin{cases}h_i(x)=0,\ i=1,2,\cdots,l,\\ g_i(x)\ge 0,\ i=1,2,\cdots,m,\end{cases}
\end{aligned}\tag{8.15}$$

若 $f(x)$ 是凸函数, $h_i(x)\,(i=1,\cdots,l)$ 是线性函数, $g_i(x)\,(i=1,\cdots,m)$ 是凹函数 (即 $-g_i(x)$ 是凸函数), 那么上述约束优化问题称为凸优化问题.

定理 41 设 (x^*,μ^*,λ^*) 是凸优化问题 (8.15) 的 KT 点, 则 x^* 必为该问题的全局极小点.

证 因对于凸优化问题, 其拉格朗日函数

$$L(x,\mu^*,\lambda^*)=f(x)-\sum_{i=1}^{l}\mu_i^*h_i(x)-\sum_{i=1}^{m}\lambda_i^*g_i(x)$$

关于 x 是凸函数. 故对于每一个可行点 x, 我们有

$$\begin{aligned}
f(x) &\geq f(x)-\sum_{i=1}^{l}\mu_i^*h_i(x)-\sum_{i=1}^{m}\lambda_i^*g_i(x)\\
&= L(x,\mu^*,\lambda^*)\\
&\geq L(x^*,\mu^*,\lambda^*)+\nabla_x L(x^*,\mu^*,\lambda^*)^T(x-x^*)\\
&= L(x^*,\mu^*,\lambda^*)\ =\ f(x^*).
\end{aligned}$$

故 x^* 为问题的全局极小点. □

8.4　鞍点和对偶问题

本节介绍约束优化问题的鞍点和对偶等有关概念. 首先给出鞍点的定义.

定义 19 对约束优化问题 (8.11), 若存在 x^* 和 (μ^*,λ^*), 其中 $\lambda^*\geq 0$, 满足

$$L(x^*,\mu,\lambda)\leq L(x^*,\mu^*,\lambda^*)\leq L(x,\mu^*,\lambda^*),\quad \forall\,(x,\mu,\lambda)\in\mathbb{R}^n\times\mathbb{R}^l\times\mathbb{R}^m_+, \tag{8.16}$$

则称 (x^*,μ^*,λ^*) 为约束优化问题 (8.11) 的拉格朗日函数的鞍点, 通常简称 x^* 为问题 (8.11) 的鞍点.

下面的定理表明, 鞍点 x^* 不仅是 KT 点, 而且是全局极小点.

定理 42 设 (x^*,μ^*,λ^*) 是约束优化问题 (8.11) 的鞍点. 则 (x^*,μ^*,λ^*) 不仅是问题 (8.11) KT 点, 而且是它的全局极小点.

证 由鞍点的定义 19 可知 x^* 是

$$\min_{x\in\mathbb{R}^n} L(x,\mu^*,\lambda^*)$$

的全局极小点. 由无约束优化问题的最优性条件可得 $\nabla_x L(x^*,\mu^*,\lambda^*)=0$, 这就证明了约束优化问题 (8.11) KT 条件的第一个式子.

另一方面, 再由鞍点的定义可知 (μ^*,λ^*) 是

$$\max_{\lambda_i\geq 0\,(i\in I),\mu\in\mathbb{R}^l} L(x^*,\mu,\lambda)$$

的全局极大点, 等价地, (μ^*,λ^*) 是

$$\min_{\lambda_i\geq 0\,(i\in I),\mu\in\mathbb{R}^l} -L(x^*,\mu,\lambda)$$

的全局极小点. 那么由定理 39 可知, 存在乘子向量 $\omega^*=(\omega_1^*,\cdots,\omega_m^*)^T$ $(\omega_i^*\geq 0, i=1,\cdots,m)$ 使得

$$\begin{cases} h_i(x^*)=0, & i\in E \\ g_i(x^*)=\omega_i^*\geq 0,\ \lambda_i^*\geq 0,\ \omega_i^*\lambda_i^*=0, & i\in I. \end{cases}$$

从而 x^* 是问题 (8.11) 的可行点, 且 $\lambda_i^* g_i(x^*)=0,\ i\in I$, 故 (x^*,μ^*,λ^*) 满足 (8.11) 的 KT 条件, 即为 KT 点.

进一步, 由鞍点的定义, 对于问题 (8.11) 的任意可行点 x, 我们有

$$L(x^*,\mu^*,\lambda^*)\leq L(x,\mu^*,\lambda^*),$$

即

$$f(x^*)\leq f(x)-\sum_{i=1}^{l}\mu_i^* h_i(x)-\sum_{i=1}^{m}\lambda_i^* g_i(x)\leq f(x),$$

从而 x^* 是问题 (8.11) 的全局极小点. □

上述定理说明, 鞍点一定是 KT 点, 但反之不一定成立. 然而, 对于凸优化问题, KT 点、鞍点和全局极小点三者是等价的. 我们有

定理 43 设 (x^*,μ^*,λ^*) 是凸优化问题的 KT 点, 则 (x^*,μ^*,λ^*) 为对应的拉格朗日函数的鞍点, 同时 x^* 也是该凸优化问题的全局极小点.

证 注意到对于凸优化问题, 拉格朗日函数

$$L(x,\mu^*,\lambda^*)=f(x)-\sum_{i=1}^{l}\mu_i^*h_i(x)-\sum_{i=1}^{m}\lambda_i^*g_i(x)$$

关于 x 是凸函数, 故由凸函数的性质 (定理 4), 有

$$\begin{aligned}L(x,\mu^*,\lambda^*) &\geq L(x^*,\mu^*,\lambda^*)+\nabla_x L(x^*,\mu^*,\lambda^*)^T(x-x^*)\\ &= L(x^*,\mu^*,\lambda^*),\end{aligned}$$

即 $L(x^*,\mu^*,\lambda^*)\leq L(x,\mu^*,\lambda^*)$. 另一方面, 对于任意的 $(\lambda,\mu)\in\mathbb{R}_+^m\times\mathbb{R}^l$, 有

$$\begin{aligned}L(x^*,\mu,\lambda)-L(x^*,\mu^*,\lambda^*) &= -\sum_{i=1}^{l}(\mu_i-\mu_i^*)h_i(x^*)-\sum_{i=1}^{m}(\lambda_i-\lambda_i^*)g_i(x^*)\\ &= -\sum_{i=1}^{m}\lambda_i g_i(x^*)\leq 0,\end{aligned}$$

即 $L(x^*,\mu,\lambda)\leq L(x^*,\mu^*,\lambda^*)$, 故 (x^*,μ^*,λ^*) 为鞍点, 同时 x^* 也是凸优化问题的全局极小点. □

下面讨论约束优化问题的对偶问题. 对于约束优化问题 (8.11), 我们引入如下记号:

$$G(x)=(g_1(x),\cdots,g_m(x))^T,\quad H(x)=(h_1(x),\cdots,h_l(x))^T.$$

令 $y\in\mathbb{R}^m$, $z\in\mathbb{R}^l$, 定义函数

$$L(x,y,z)=f(x)-G(x)^Ty-H(x)^Tz$$

及

$$\theta(y,z)=\inf_{x\in\mathbb{R}^n}L(x,y,z).$$

易知 $\theta(y,z)$ 关于 (y,z) 是凹函数.

约束问题 (8.11) 的拉格朗日对偶定义为:

$$\begin{cases}\max\ \theta(y,z),\\ \text{s.t.}\quad y\in\mathbb{R}_+^m,\ z\in\mathbb{R}^l.\end{cases}\tag{8.17}$$

约束问题 (8.11) 的 Wolfe 对偶定义为:

$$
\begin{cases}
\max\ L(x,y,z),\\
\text{s.t.}\ \ \nabla_x L(x,y,z)=0,\\
\qquad y\in\mathbb{R}^m_+,\ z\in\mathbb{R}^l.
\end{cases}
\tag{8.18}
$$

注 4 上面的两种对偶在某种意义上是一致的. 事实上, 对于拉格朗日对偶, 由于目标函数 $\theta(y,z)$ 本身就是拉格朗日函数关于 x 的极小值, 所以显然有 $\nabla_x L(x,y,z)=0$ 成立. 将其合并到拉格朗日对偶的约束当中, 就得到了所谓的 Wolfe 对偶.

对于线性规划问题和凸二次规划问题, 拉格朗日对偶 (8.17) 将会有更为明晰的形式. 我们看下面例子.

例 23 设有线性规划问题

$$
\begin{cases}
\min\ c^Tx,\\
\text{s.t.}\ \ Ax=b,\\
\qquad x\geq 0.
\end{cases}
\tag{8.19}
$$

试写出它的拉格朗日对偶.

解 其拉格朗日函数为

$$L(x,y,z)=c^Tx-y^Tx-z^T(Ax-b).$$

对上述函数关于 x 求极小. 令

$$\nabla_x L(x,y,z)=c-y-A^Tz=0.$$

将其代入拉格朗日函数得

$$
\begin{aligned}
\theta(y,z) &= \inf_{x\in\mathbb{R}^n} L(x,y,z)\\
&= \inf_{x\in\mathbb{R}^n}\{(c-y-A^Tz)^Tx+z^Tb\}\\
&= z^Tb\ =\ b^Tz.
\end{aligned}
$$

注意到 $y = c - A^T z \geq 0$, 于是有

$$\begin{cases} \max\ b^T z, \\ \text{s.t.}\quad A^T z \leq c. \end{cases} \tag{8.20}$$

这就是线性规划问题 (8.19) 的对偶规划.

例 24 设二次规划问题

$$\begin{cases} \min\ \frac{1}{2}x^T Bx + c^T x, \\ \text{s.t.}\quad Ax \leq b, \end{cases} \tag{8.21}$$

其中 $B \in \mathbb{R}^{n\times n}$ 对称正定, $A \in \mathbb{R}^{m\times n}$. 试写出二次规划问题 (8.21) 的对偶规划.

解 首先写出该问题的拉格朗日函数为

$$L(x,y) = \frac{1}{2}x^T Bx + c^T x - y^T(b - Ax).$$

对上述函数关于 x 求极小. 由于 B 对称正定, 故函数 $L(x,y)$ 关于 x 为凸函数. 令

$$\nabla_x L(x,y,z) = Bx + c + A^T y = 0,$$

解得 $x = -B^{-1}(c + A^T y)$. 将其代入拉格朗日函数得

$$\begin{aligned} \theta(y) &= \inf_{x\in\mathbb{R}^n} L(x,y) \\ &= \inf_{x\in\mathbb{R}^n} \left\{(Bx + c + A^T y)^T x - y^T b - \frac{1}{2}x^T Bx\right\} \\ &= -b^T y - \frac{1}{2}[-B^{-1}(c + A^T y)]^T B[-B^{-1}(c + A^T y)] \\ &= -b^T y - \frac{1}{2}(c^T + y^T A)B^{-1}(c + A^T y) \\ &= (-b - AB^{-1}c)^T y - \frac{1}{2}y^T(AB^{-1}A^T)y - \frac{1}{2}c^T B^{-1}c. \end{aligned}$$

令

$$d = -b - AB^{-1}c, \quad D = -AB^{-1}A^T,$$

则有

$$\theta(y) = \frac{1}{2}y^T Dy + d^T y - \frac{1}{2}c^T B^{-1}c.$$

注意到乘子向量 $y \geq 0$, 因此二次规划问题 (8.21) 的拉格朗日对偶是

$$\begin{cases} \max \ \dfrac{1}{2}y^T D y + d^T y - \dfrac{1}{2}c^T B^{-1} c, \\ \text{s.t.} \quad y \leq 0. \end{cases}$$

最后, 我们讨论一下原问题与对偶问题的目标函数值之间的关系.

定理 44（弱对偶定理） 设 $\bar{x}$ 和 $(\bar{y}, \bar{z})$ 分别是原问题 (8.11) 和对偶问题 (8.17) 的可行解. 则有 $\theta(\bar{y}, \bar{z}) \leq f(\bar{x})$.

证 因 $\bar{x}$ 和 $(\bar{y}, \bar{z})$ 分别是原问题 (8.11) 和对偶问题 (8.17) 的可行解, 故

$$\begin{aligned} \theta(\bar{y}, \bar{z}) &= \inf_{x \in \mathbb{R}^n} \left\{ f(x) - \bar{y}^T G(x) - \bar{z}^T H(x) \right\} \\ &\leq f(\bar{x}) - \bar{y}^T G(\bar{x}) - \bar{z}^T H(\bar{x}) \ = \ f(\bar{x}). \end{aligned}$$

定理得证. □

习 题 8

1. 验证 $\bar{x} = (2, 1)^T$ 是否为下列最优化问题的 K-T 点：

$$\begin{aligned} \min \quad & f(x) = (x_1 - 3)^2 + (x_2 - 2)^2, \\ \text{s.t.} \quad & x_1^2 + x_2^2 \leq 5, \\ & x_1 + 2x_2 = 4, \\ & x_1, x_2 \geq 0. \end{aligned}$$

2. 对于最优化问题：

$$\begin{aligned} \min \quad & f(x) = 4x_1 - 3x_2, \\ \text{s.t.} \quad & -(x_1 - 3)^2 + x_2 + 1 \geq 0, \\ & 4 - x_1 - x_2 \geq 0, \\ & x_2 + 7 \geq 0. \end{aligned}$$

求满足 K-T 条件的点.

3. 写出下列优化问题的 KT 条件, 其中 $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $H \in \mathbb{R}^{n \times n}$.

(1) $\min \quad c^T x$

$\text{s.t.} \quad Ax = b,$

$x \geq 0,$

(2) $\min \quad \frac{1}{2}x^T H x + d^T x$

$\text{s.t.} \quad Ax \leq b,$

(3) $\min \quad f(x)$

$\text{s.t.} \quad h_j(x) = 0,\ j = 1, \cdots, l$

$g_i(x) \leq 0,\ i = 1, \cdots, m_1$

$g_i(x) \geq 0,\ i = m_1 + 1, \cdots, m$

4. 考虑无约束优化问题 $\min f(x)$, 设 x_k 为当前迭代点, $\nabla f(x_k) \neq 0$ 为确定 $f(x)$ 在x_k处的一个下降方向, 分别按以下两种方法构造子问题:

(1) $\min \quad \nabla f(x_k)^T d$

$\text{s.t.} \quad d^T d = 1,$

(2) $\min \quad \nabla f(x_k)^T d,$

$\text{s.t.} \quad d^T H_k d = 1,$

其中 H_k 对称正定, 试用 KT 法求解 (1), (2). 所谓 KT 法是指: 先求问题的 KT 点, 然后用最优性条件、凸分析或其它方法判别这些点是否为局部或全局最优点.

5. 利用 KT 条件推出线性规划

$$\begin{aligned} \min \quad & z = c^T x, \\ \text{s.t.} \quad & Ax \leq b, \\ & x \geq 0 \end{aligned}$$

的最优性条件.

6. 设二次规划

$$\begin{aligned} \min \quad & f(x) = \tfrac{1}{2}x^T H x + c^T x, \\ \text{s.t.} \quad & Ax = b, \end{aligned}$$

其中 H 是 n 阶对称正定阵, 矩阵 A 行满秩, 求其最优解并说明解的唯一性.

7. 设 x^* 是问题

$$\begin{aligned} \min \quad & z = c^T x \\ \text{s.t.} \quad & h(x) = 0,\ x \geq 0 \end{aligned}$$

的可行点, 且存在 u^*, 使得

$$c + \nabla h(x^*)u^* \geq 0,\ (x^*)^T(c + \nabla h(x^*)u^*) = 0,$$

其中 $h(x) = (h_1(x), \cdots, h_l(x))^T$. 证明 x^* 是问题的 KT 点.

8. 考虑 Wolfe 问题

$$\begin{aligned} \min \quad & f(x) = \frac{4}{3}(x_1^2 - x_1x_2 + x_2^2)^{\frac{3}{4}} - x_3, \\ \text{s.t.} \quad & x_1, x_2, x_3 \geq 0, \ x_3 \leq 2. \end{aligned}$$

(1) 证明目标函数 $f(x)$ 是可行集 S 上的凸函数, 其中

$$S = \{(x_1, x_2, x_3)^T t | x_1, x_2, x_3 \geq 0, x_3 \leq 2\}.$$

(2) 试用 KT 条件求此问题的全局极小点.

9. 给定线性约束优化问题

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} \frac{c_i}{x_i}, \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_i x_i = b, \end{aligned}$$

其中 $a_i, c_i (i = 1, \cdots, n)$, b 是正常数. 证明目标函数的全局极小值为:

$$f(x^*) = \frac{1}{b}\left[\sum_{i=1}^{n} \sqrt{a_i c_i}\right]^2.$$

第九章　罚函数法

从本章开始, 我们讨论约束优化问题的求解方法. 首先介绍求解约束优化问题的经典算法 — 罚函数法. 其基本思想是: 根据约束条件的特点将其转化为某种惩罚函数加到目标函数中去, 从而将约束优化问题转化为一系列的无约束优化问题来求解. 本章主要介绍外罚函数法、内点法和乘子法.

9.1　外罚函数法

我们首先通过一个简单的例子来说明罚函数的构造.

例 25　*求解约束优化问题:*

$$\begin{aligned} &\min\ f(x) = (x_1-1)^2 + (x_2-1)^2, \\ &\text{s.t.}\ \ x_1 + x_2 = 1. \end{aligned}$$

解　由等式约束得 $x_2 = 1 - x_1$, 代入目标函数得到一个无约束的单变量极小化问题

$$\min\ \phi(x_1) = (x_1-1)^2 + x_1^2,$$

其全局极小点为 $x_1^* = 0.5$, 从而得到原问题的全局极小点为 $x^* = (0.5, 0.5)^T$. 现在要使构造的罚函数 $\bar{P}(x)$ 满足

$$\bar{P}(x) \begin{cases} = 0, & x_1 + x_2 - 1 = 0, \\ > 0, & x_1 + x_2 - 1 \neq 0, \end{cases}$$

只要令 $\bar{P}(x) = (x_1 + x_2 - 1)^2$ 即可. 现在考察目标函数和上述罚函数的组合

$$\begin{aligned} P(x,\sigma) &= f(x) + \sigma \bar{P}(x) \\ &= (x_1-1)^2 + (x_2-1)^2 + \sigma(x_1+x_2-1)^2, \end{aligned}$$

其中 $\sigma > 0$ 是充分大的正数, 称为罚参数或罚因子. 求这个组合函数的极小点. 由

$$\frac{\partial P(x,\sigma)}{\partial x_1} = \frac{\partial P(x,\sigma)}{\partial x_2} = 0,$$

得

$$\begin{cases} (1+\sigma)x_1 + \sigma x_2 = 1+\sigma, \\ \sigma x_1 + (1+\sigma)x_2 = 1+\sigma. \end{cases}$$

求解上述方程组得

$$x_1(\sigma) = x_2(\sigma) = \frac{\sigma+1}{2\sigma+1}.$$

令 $\sigma \to +\infty$, 有

$$\big(x_1(\sigma), x_2(\sigma)\big)^T \to \left(\frac{1}{2}, \frac{1}{2}\right)^T = x^*.$$

这样, 我们就从无约束优化问题极小点的极限得到了原问题的极小点. 下面我们将这种思想方法推广到一般约束的优化问题. 考虑

$$\begin{cases} \min\ f(x), & x \in \mathbb{R}^n, \\ \text{s.t.}\ \ h_i(x) = 0, & i \in E = \{1, \cdots, l\}, \\ \qquad\ g_i(x) \geq 0, & i \in I = \{1, \cdots, m\}. \end{cases} \tag{9.1}$$

记可行域为 $\mathcal{D} = \{x \in \mathbb{R}^n \,|\, h_i(x) = 0\,(i \in E),\ g_i(x) \geq 0\,(i \in I)\}$. 构造罚函数

$$\bar{P}(x) = \sum_{i=1}^{l} h_i^2(x) + \sum_{i=1}^{m} [\min\{0, g_i(x)\}]^2 \tag{9.2}$$

和增广目标函数

$$P(x,\sigma) = f(x) + \sigma \bar{P}(x), \tag{9.3}$$

其中 $\sigma > 0$ 是罚参数或罚因子. 不难发现, 当 $x \in \mathcal{D}$ 时, 即 x 为可行点时, $P(x,\sigma) = f(x)$, 此时目标函数没有受到额外惩罚; 而当 $x \notin \mathcal{D}$ 时, 即 x 为不可行点时, $P(x,\sigma) > f(x)$, 时目标函数受到了额外的惩罚. $\sigma > 0$ 越大, 受到的惩罚越重. 当 $\sigma > 0$ 充分大时, 要使 $P(x,\sigma)$ 达到极小, 罚函数 $\bar{P}(x)$ 应充分小才可以. 从而 $P(x,\sigma)$ 的极小点充分逼近可行域 $\mathcal{D}$, 而其极小值自然充分逼近 $f(x)$ 在 $\mathcal{D}$ 上的极小值. 这样求解一般约束优化问题 (9.1) 就可以化为求解一系列无约束的优化问题

$$\min\ P(x, \sigma_k),$$

其中 $\{\sigma_k\}$ 是正数序列, 且 $\sigma_k \to +\infty$.

从例 25 可以看出, 当 $\sigma \to \infty$ 时, $P(x,\sigma)$ 的极小点 $x(\sigma) \to x^*$, 但

$$x_1(\sigma) + x_2(\sigma) - 1 = \frac{2\sigma+2}{2\sigma+1} - 1 = \frac{1}{2\sigma+1} \neq 0,$$

即 $x(\sigma) \notin \mathcal{D}$, 也就是说 $x(\sigma)$ 是从可行域的外部趋于 x^* 的. 因此上述的罚函数法也称为外罚函数法 (或外点法).

下面给出外罚函数法的详细算法步骤.

算法 21 (外罚函数法)

步 0　给定初始点 $x_0 \in \mathbb{R}^n$, 终止误差 $0 \le \varepsilon \ll 1$. $\sigma_1 > 0$, $\gamma > 1$. 令 $k := 1$.

步 1　以 x_{k-1} 为初始点求解子问题

$$\min_{x \in \mathbb{R}^n} P(x,\sigma_k) = f(x) + \sigma_k \bar{P}(x). \tag{9.4}$$

令其极小点为 x_k.

步 2　若 $\sigma_k \bar{P}(x_k) \le \varepsilon$, 停算, 输出 $x^* \approx x_k$ 作为近似极小点.

步 3　令 $\sigma_{k+1} := \gamma\sigma_k$, $k := k+1$, 转步 1.

注　由上述算法可知, 外罚函数法结构简单, 可以直接调用无约束优化算法的通用程序, 因而容易编程实现. 缺点是: (1) x_k 往往不是可行点, 这对于某些实际问题是难以接受的; (2) 罚参数 σ_k 的选取比较困难, 取的过小, 可能起不到 “惩罚” 的作用, 而取得过大则可能造成 $P(x,\sigma_k)$ 的 Hesse 阵的条件数很大, 从而带来数值技术上的困难; (3) 注意到 $\bar{P}(x)$ 一般是不可微的, 因而难以直接使用利用导数的优化算法, 从而收敛速度缓慢.

下面讨论算法 21 的收敛性. 我们首先证明下面的引理.

引理 14　设 $\{x_k\}$ 是由算法 21 产生的迭代序列. 若 x_k 是子问题 (9.4) 的全局极小点, 则有下述结论成立:

$$P(x_{k+1},\sigma_{k+1}) \ge P(x_k,\sigma_k), \tag{9.5}$$

$$\bar{P}(x_{k+1}) \le \bar{P}(x_k), \tag{9.6}$$

$$f(x_{k+1}) \ge f(x_k). \tag{9.7}$$

证 (1) 注意到 $\sigma_{k+1} \geq \sigma_k > 0$, 因此有

$$\begin{aligned} P(x_{k+1},\sigma_{k+1}) &= f(x_{k+1}) + \sigma_{k+1}\bar{P}(x_{k+1}), \\ &\geq f(x_{k+1}) + \sigma_k\bar{P}(x_{k+1}), \\ &= P(x_{k+1},\sigma_k) \geq P(x_k,\sigma_k), \end{aligned}$$

即 (9.5) 成立. 由题设 x_k, x_{k+1} 分别是 $P(x,\sigma_k)$ 和 $P(x,\sigma_{k+1})$ 的全局极小点, 故有

$$f(x_{k+1}) + \sigma_k\bar{P}(x_{k+1}) \geq f(x_k) + \sigma_k\bar{P}(x_k), \tag{9.8}$$

$$f(x_k) + \sigma_{k+1}\bar{P}(x_k) \geq f(x_{k+1}) + \sigma_{k+1}\bar{P}(x_{k+1}). \tag{9.9}$$

上面两式相加并整理可得

$$(\sigma_{k+1} - \sigma_k)\bar{P}(x_k) \geq (\sigma_{k+1} - \sigma_k)\bar{P}(x_{k+1}),$$

即

$$(\sigma_{k+1} - \sigma_k)[\bar{P}(x_k) - \bar{P}(x_{k+1})] \geq 0,$$

从而必有 $\bar{P}(x_k) - \bar{P}(x_{k+1}) \geq 0$, 即 (9.6) 成立. 最后, 由 (9.8) 立即可得

$$f(x_{k+1}) - f(x_k) \geq \sigma_k[\bar{P}(x_k) - \bar{P}(x_{k+1}) \geq 0.$$

证毕. □

下面我们给出算法 21 的收敛性定理.

定理 45 设 $\{x_k\}$ 和 $\{\sigma_k\}$ 是由算法 21 产生的序列, x^* 是约束优化问题 (9.1) 的全局极小点. 若 x_k 为无约束子问题 (9.4) 的全局极小点, 且罚参数 $\sigma_k \to +\infty$, 则 $\{x_k\}$ 的任一聚点 $\bar{x}$ 都是 (9.1) 的全局极小点.

证 设 x^∞ 是序列 $\{x_k\}$ 的一个聚点, 不失一般性, 设 $x_k \to x^\infty$, $k \to +\infty$. 由题设 x^* 是原问题的全局极小点, 因而必为可行点, 故有 $\bar{P}(x^*) = 0$. 下面分两步证明本定理的结论.

(1) 先证 x^∞ 是原问题的可行点, 亦即 $\bar{P}(x^\infty) = 0$. 事实上, 由引理 14 知 $\{P(x_k,\sigma_k)\}$ 是单调递增有上界的序列, 因此极限存在, 设为 P^∞. 此外, 注意到 $\{f(x_k)\})$ 也是单调递增的, 且

$$f(x_k) \leq P(x_k,\sigma_k) \leq f(x^*),$$

即序列 $\{f(x_k)\}$ 也收敛, 记其极限为 f^∞. 于是有

$$\lim_{k\to\infty}\sigma_k\bar{P}(x_k)=\lim_{k\to\infty}\big[P(x_k,\sigma_k)-f(x_k)\big]=P^\infty-f^\infty.$$

因 $\sigma_k\to+\infty$, 故必有 $\lim\limits_{k\to\infty}\bar{P}(x_k)=0$. 由 $\bar{P}$ 的连续性知 $\bar{P}(x^\infty)=0$, 即 x^∞ 是可行点.

(2) 再证 x^∞ 是全局极小点, 亦即 $f(x^\infty)=f(x^*)$. 由 $f(x)$ 的连续性及 $x_k\to x^\infty$ 可知

$$f(x^\infty)=\lim_{k\to\infty}f(x_k)\le f(x^*).$$

注意到 x^* 是问题的全局极小点, 故显然有 $f(x^*)\le f(x^\infty)$. 从而 $f(x^\infty)=f(x^*)$, 从而 x^∞ 为原问题的全局极小点. 证毕. □

注 上述定理要求算法的每一迭代步求解子问题得到的 x_k 必须是无约束问题 $\min P(x,\sigma_k)$ 的全局极小点. 这一点在实际计算中是很难操作的, 因为求无约束优化问题全局极小点至今仍然是一个很困难的问题. 故算法 21 (外罚函数法) 经常遇到迭代失败的情形是很正常的. 此外, 算法 21 之所以选用 $\sigma_k\bar{P}(x_k)\le\varepsilon$ 作为终止条件, 是因为

$$\lim_{\infty}\sigma_k\bar{P}(x_k)=\lim_{k\to\infty}[P(x_k,\sigma_k)-f(x_k)]P^\infty=-f^\infty=0$$

的缘故.

9.2　内点法

1. 不等式约束问题的内点法

内点法一般只适用于不等式约束的优化问题:

$$\begin{cases}\min\ f(x), & x\in\mathbb{R}^n,\\ \text{s.t.}\ \ g_i(x)\ge 0, & i=1,\cdots,m.\end{cases}\tag{9.10}$$

记可行域 $\mathcal{D}=\{x\in\mathbb{R}^n:g_i(x)\ge 0,\ i=1,\cdots,m\}$. 内点法也属于罚方法的范畴, 其基本思想是保持每一个迭代点 x_k 是可行域 $\mathcal{D}$ 的内点, 可行域的边界被筑起一道很高的“围墙”作为障碍, 当迭代点靠近边界时, 增广目标函数值骤然增大, 以

示“惩罚”, 并阻止迭代点穿越边界. 因此, 内点法也称为内罚函数法或障碍函数法, 它只是用于可行域的内点集非空的情形, 即

$$\mathcal{D}_0 = \{x \in \mathbb{R}^n : g_i(x) > 0,\ i = 1, \cdots, m\} \neq \emptyset.$$

类似于外罚函数法, 我们需要构造如下的增广目标函数

$$H(x, \tau) = f(x) + \tau \bar{H}(x),$$

其中 $\bar{H}(x)$ 是障碍函数, 它需要满足如下性质: 当 x 在 $\mathcal{D}_0$ 趋向于边界时, 至少有一个 $g_i(x)$ 趋向于 0, 而 $\bar{H}(x)$ 要趋向于无穷大. 因此可以取约束函数的倒数之和为障碍函数可满足要求, 即

$$\bar{H}(x) = \sum_{i=1}^{m} \frac{1}{g_i(x)}, \tag{9.11}$$

或者取对数障碍函数

$$\bar{H}(x) = -\sum_{i=1}^{m} \ln[g_i(x)]. \tag{9.12}$$

参数 $\tau > 0$ 称为罚因子或罚参数. 这样, 当 x 在 $\mathcal{D}_0$ 中时, $\bar{H}(x) > 0$ 是有限的; 当 x 接近边界时, $\bar{H}(x) \to +\infty$, 从而增广目标函数的值也趋向于无穷大, 因此得到了严重的“惩罚”.

由于约束优化问题的极小点一般在可行域的边界上达到, 因此与外罚函数法中的罚因子 $\sigma_k \to +\infty$ 相反, 内点法中的罚因子则要求 $\tau_k \to 0$. 于是, 求解问题 (9.10) 就可以转化为求解序列无约束优化子问题:

$$\min\ H(x, \tau_k) = f(x) + \tau_k \bar{H}(x). \tag{9.13}$$

现在我们来看一个简单的例子.

例 26 用内点法求解下面的优化问题:

$$\begin{cases} \min & 2x_1 + 3x_2, \\ \text{s.t.} & 1 - 2x_1^2 - x_2^2 \geq 0. \end{cases}$$

解 给出增广目标函数为

$$H(x, \tau) = 2x_1 + 3x_2 - \tau \ln(1 - 2x_1^2 - x_2^2).$$

令

$$\frac{\partial H}{\partial x_1} = 2 + \frac{4\tau x_1}{1-2x_1^2-x_2^2} = 0, \tag{9.14}$$

$$\frac{\partial H}{\partial x_2} = 3 + \frac{2\tau x_2}{1-2x_1^2-x_2^2} = 0. \tag{9.15}$$

由 (9.14)-(9.15) 可得 $x_2 = 3x_1$. 代入 (9.14) 得

$$0 = 1 + \frac{2\tau x_1}{1-2x_1^2-(3x_1)^2} = 1 + \frac{2\tau x_1}{1-11x_1^2},$$

即

$$11x_1^2 - 2\tau x_1 - 1 = 0.$$

解得

$$x_1(\tau) = \frac{\tau \pm \sqrt{\tau^2+11}}{11} \ \to\ \pm\frac{1}{\sqrt{11}},\ (\tau \to 0).$$

自然有

$$x_2(\tau) = 3x_1(\tau) \ \to\ \pm\frac{3}{\sqrt{11}},\ (\tau \to 0).$$

注意到目标函数的表达式, 可得所求问题的全局极小点和全局极小值分别为

$$x^* = \Big(-\frac{1}{\sqrt{11}}, -\frac{3}{\sqrt{11}}\Big)^T, \quad f(x^*) = 2x_1^* + 3x_2^* = -\sqrt{11}.$$

上面的问题比较简单, 可以将子问题 $\min H(x,\tau)$ 最优解的解析表达式求出来, 然后对罚参数 $\tau \to 0$ 取极限而得到原问题的极小点. 一般来说, 对于较复杂的问题, 只能用数值方法来求子问题的近似全局极小点.

下面给出内点法的详细算法步骤.

算法 22 (内点法)

步 0　给定初始点 $x_0 \in \mathcal{D}_0$, 终止误差 $0 \le \varepsilon \ll 1$. $\tau_1 > 0$, $\varrho \in (0,1)$. 令 $k := 1$.

步 1　以 x_{k-1} 为初始点求解无约束子问题 (9.13), 得极小点 x_k.

步 2　若 $\tau_k \bar{H}(x_k) \le \varepsilon$, 停算, 输出 $x^* \approx x_k$ 作为近似极小点.

步 3　令 $\tau_{k+1} := \varrho\tau_k$, $k := k+1$, 转步 1.

注　由上述算法可以看出, 内点法的优点是结构简单, 适应性强. 但是随着迭代过程的进行, 罚参数 τ_k 将变得越来越小, 趋向于零, 使得增广目标函数的病态

性越来越严重, 这给无约束子问题的求解带来了数值实现上的困难, 以致导致迭代的失败. 此外, 内点法的初始点 x_0 要求是一个严格的可行点, 一般来说这也是比较麻烦甚至困难的.

下面我们来考虑内点法的收敛性. 先看下面的引理.

引理 15 设序列 $\{x_k\}$ 由算法 22 产生, 且每个 x_k 都是无约束子问题 (9.13) 的全局极小点. 那么增广目标函数序列 $\{H(x_k,\tau_k)\}$ 是单调下降的, 即

$$H(x_{k+1},\tau_{k+1}) \le H(x_k,\tau_k).$$

证 注意到 x_{k+1} 是 $H(x,\tau_{k+1})$ 的全局极小点, 且由算法有 $\tau_{k+1} \le \tau_k$, 故

$$\begin{aligned} H(x_{k+1},\tau_{k+1}) &= f(x_{k+1}) + \tau_{k+1}\bar{H}(x_{k+1}) \\ &\le f(x_k) + \tau_{k+1}\bar{H}(x_k) \\ &\le f(x_k) + \tau_k\bar{H}(x_k) \\ &= H(x_k,\tau_k). \end{aligned}$$

证毕. □

下面的定理给出了算法 22 的收敛性.

定理 46 设 $f(x)$ 在 $\mathcal{D}$ 上存在全局极小点 x^* 且内点集 $\mathcal{D}_0 \ne \emptyset$. $\{(x_k,\tau_k)\}$ 是由算法 22 产生的序列. 若 x_k 是 $H(x,\tau_k)$ 的全局极小点且 $\{\tau_k\}\downarrow 0$, 那么 $\{x_k\}$ 的任一聚点 $\bar{x}$ 都是问题 (9.10) 的全局极小点.

证 由定理的条件, 有 $x_k \in \mathcal{D}_0 \subset \mathcal{D}$ 且 x^* 是 $f(x)$ 在 $\mathcal{D}$ 上的全局极小点. 故有 $f(x^*) \le f(x_k) \le H(x_k,\tau_k)$, 即序列 $\{H(x_k,\tau_k)\}$ 有下界, 于是由引理 15 知 $\lim\limits_{k\to\infty} H(x_k,\tau_k)$ 存在, 不妨记为 H^*.

下面证明 $H^* = f(x^*)$. 事实上, 显然有 $H^* \ge f(x^*)$. 以下只需证明 $H^* \le f(x^*)$. 由 $f(x)$ 的连续性可知, 对于任意的 $\varepsilon > 0$, 存在 $\delta > 0$, 使得满足 $\|\bar{x} - x^*\| \le \delta$ 的 $\bar{x} \in \mathcal{D}_0$, 有

$$f(\bar{x}) - f(x^*) < \varepsilon.$$

又因 $\{\tau_k\}\downarrow 0$, 故对于上述的 ε 和 $\bar{x}$, 存在正数 k_0, 使得当 $k \ge k_0$ 时, 有

$$\tau_k\bar{H}(x_k) \le \varepsilon.$$

注意到 x_k 是 $H(x,\tau_k)$ 的全局极小点, 即有

$$H(x_k,\tau_k) \le H(\bar{x},\tau_k).$$

这样, 我们有

$$\begin{aligned} H(x_k,\tau_k) - f(x^*) &\le H(\bar{x},\tau_k) - f(x^*) \\ &= [f(\bar{x}) - f(x^*)] + \tau_k \bar{H}(\bar{x}) \\ &< \varepsilon + \varepsilon = 2\varepsilon. \end{aligned}$$

令 $\varepsilon \to 0^+$, 并对上式取极限, 即得 $H^* \le f(x^*)$, 从而 $H^* = f(x^*)$. 最后, 对不等式

$$f(x^*) \le f(x_k) \le H(x_k,\tau_k)$$

取极限得 $\lim\limits_{k\to\infty} f(x_k) = f(x^*)$. □

2. 一般约束问题的内点法

现在, 我们考虑一般约束优化问题

$$\begin{cases} \min\ f(x), & x \in \mathbb{R}^n, \\ \text{s.t.}\ \ h_i(x) = 0, & i = 1,\cdots,l, \\ \qquad g_i(x) \ge 0, & i = 1,\cdots,m \end{cases} \tag{9.16}$$

内点法特征的罚函数方法. 途径之一是对于等式约束利用“外罚函数”的思想, 而对于不等式约束则利用“障碍函数”的思想, 构造出所谓混合增广目标函数

$$H(x,\mu) = f(x) + \frac{1}{2\mu}\sum_{i=1}^{l} h_i^2(x) + \mu\sum_{i=1}^{m}\frac{1}{g_i(x)}, \tag{9.17}$$

或

$$H(x,\mu) = f(x) + \frac{1}{2\mu}\sum_{i=1}^{l} h_i^2(x) - \mu\sum_{i=1}^{m}\ln[g_i(x)]. \tag{9.18}$$

于是可以类似于内点法或外罚函数法的算法框架建立起相应的算法. 但由此建立的算法的初始点的选取仍然是个困难的问题.

另一种途径是, 引入松弛变量 y_i, $i=1,\cdots,m$, 将问题等价地转化为

$$\begin{cases} \min\ f(x), & x\in\mathbb{R}^n, \\ \text{s.t.}\ \ h_i(x)=0, & i=1,\cdots,l, \\ \qquad g_i(x)-y_i=0, & i=1,\cdots,m, \\ \qquad y_i\ge 0, & i=1,\cdots,m. \end{cases} \tag{9.19}$$

然后构造等价问题 (9.19) 的混合增广目标函数:

$$\psi(x,y,\mu)=f(x)+\frac{1}{2\mu}\sum_{i=1}^{l}h_i^2(x)+\frac{1}{2\mu}\sum_{i=m}^{l}\left[g_i(x)-y_i\right]^2+\mu\sum_{i=1}^{m}\frac{1}{y_i}, \tag{9.20}$$

或

$$\psi(x,y,\mu)=f(x)+\frac{1}{2\mu}\sum_{i=1}^{l}h_i^2(x)+\frac{1}{2\mu}\sum_{i=1}^{m}\left[g_i(x)-y_i\right]^2-\mu\sum_{i=1}^{m}\ln y_i. \tag{9.21}$$

在此基础上, 类似于前面的外罚函数法与内点法的算法框架, 可以建立起相应的求解算法. 值得说明的是, 此时, 任意的 (x,y), $y>0$ 均可作为一个合适的初始点来启动相应的迭代算法.

9.3　乘子法

乘子法是 Powell 和 Hestenes 于 1969 年针对等式约束优化问题同时独立提出的一种优化算法, 后于 1973 年经 Rockfellar 推广到求解不等式约束优化问题. 其基本思想是从原问题的拉格朗日函数出发, 再加上适当的罚函数, 从而将原问题转化为求解一系列的无约束优化子问题. 由于外罚函数法中的罚参数 $\sigma_k\to+\infty$, 因此增广目标函数变得“越来越病态”. 增广目标函数的这种病态性质是外罚函数法的主要缺点, 而这种缺陷在乘子法中由于引入拉格朗日函数及加上适当的罚函数而得以有效的克服.

1. 等式约束问题的乘子法

考虑等式约束优化问题

$$\begin{cases} \min\ f(x), \\ \text{s.t.}\ \ h(x)=0, \end{cases} \tag{9.22}$$

其中 $h(x)=(h_1(x),\cdots,h_l(x))^T$. 记可行域 $\mathcal{D}=\{x\in\mathbb{R}^n\,|\,h(x)=0\}$, 则上述问题的拉格朗日函数为

$$L(x,\lambda)=f(x)-\lambda^T h(x),$$

其中 $\lambda=(\lambda_1,\cdots,\lambda_l)^T$ 为乘子向量. 设 (x^*,λ^*) 是问题 (9.22) 的 KT 点, 则由最优性条件有

$$\nabla_x L(x^*,\lambda^*)=0,\quad \nabla_\lambda L(x^*,\lambda^*)=-h(x^*)=0.$$

此外, 不难发现, 对于任意的 $x\in\mathcal{D}$, 有

$$L(x^*,\lambda^*)=f(x^*)\le f(x)=f(x)-(\lambda^*)^T h(x)=L(x,\lambda^*).$$

上式表明, 若已知乘子向量 λ^*, 则问题 (9.22) 可等价地转化为

$$\begin{cases}\min\ L(x,\lambda^*),\\ \text{s.t.}\ \ h(x)=0.\end{cases}\tag{9.23}$$

可以考虑用外罚函数法求解问题 (9.23), 其增广目标函数为

$$\psi(x,\lambda^*,\sigma)=L(x,\lambda^*)+\frac{\sigma}{2}\|h(x)\|^2.$$

可以证明, 当 $\sigma>0$ 适当大时, x^* 是 $\psi(x,\lambda^*,\sigma)$ 的极小点. 由于乘子 λ^* 事先并不知道, 故可考虑下面的增广目标函数

$$\begin{aligned}\psi(x,\lambda,\sigma)&=L(x,\lambda)+\frac{\sigma}{2}\|h(x)\|^2\\&=f(x)-\lambda^T h(x)+\frac{\sigma}{2}\|h(x)\|^2.\end{aligned}\tag{9.24}$$

可以这样操作: 首先固定一个 $\lambda=\bar{\lambda}$, 求 $\psi(x,\bar{\lambda},\sigma)$ 的极小点 $\bar{x}$; 然后再适当改变 λ 的值, 再求新的 $\bar{x}$, 直到求得满足要求的 x^* 和 λ^* 为止. 具体地说, 我们在第 k 次迭代求无约束子问题 $\min\,\psi(x,\lambda_k,\sigma)$ 的极小点 x_k. 则由取极值的必要条件, 有

$$\nabla_x\psi(x_k,\lambda_k,\sigma)=\nabla f(x_k)-\nabla h(x_k)[\lambda_k-\sigma h(x_k)]=0.$$

而在原问题的 KT 点 (x^*,λ^*) 处有

$$\nabla f(x^*)-\nabla h(x^*)\lambda^*=0,\quad h(x^*)=0.$$

我们的目的自然希望 $\{x_k\} \to x^*$, 且 $\{\lambda_k\} \to \lambda^*$. 于是将上面两式相比较后, 可取乘子序列 $\{\lambda_k\}$ 的更新公式为

$$\lambda_{k+1} = \lambda_k - \sigma h(x_k). \tag{9.25}$$

由上式可以看出 $\{\lambda_k\}$ 收敛的充要条件是 $\{h(x_k)\} \to 0$. 下面我们证明 $h(x_k) = 0$ 也是判别 (x_k, λ_k) 是 KT 对的充要条件.

定理 47 设无约束优化问题

$$\min \psi(x, \lambda_k, \sigma) = L(x, \lambda_k) + \frac{\sigma}{2}\|h(x)\|^2 \tag{9.26}$$

的极小点为 x_k. 则 (x_k, λ_k) 是 (9.22) 的 KT 对的充要条件是 $h(x_k) = 0$.

证 必要性显然. 我们证明充分性. 因为 x_k 是 (9.26) 的极小点, 且 $h(x_k) = 0$. 则对任意的可行点 $x \in \mathcal{D} = \{x \in \mathbb{R}^n \mid h(x) = 0\}$, 有

$$f(x) = \psi(x, \lambda_k, \sigma) \ge \psi(x_k, \lambda_k, \sigma) = f(x_k),$$

即 x_k 是 (9.22) 的极小点. 另一方面, 注意到 x_k 也是 (9.26) 的稳定点, 故有

$$\nabla_x \psi(x_k, \lambda_k, \sigma) = \nabla f(x_k) - \nabla h(x_k)\lambda_k = 0.$$

上式表明 λ_k 是相应于 x_k 的拉格朗日乘子向量, 即 (x_k, λ_k) 是问题 (9.22) 的 KT 对. 证毕. □

基于上述讨论, 我们给出求解等式约束问题 (9.22) 的乘子法的详细步骤如下 (由于该算法是由 Powell 和 Hestenes 首先独立提出来的, 因此也称之为 PH 算法):

算法 23 PH **算法**

步 0 选取初始值. 给定 $x_0 \in \mathbb{R}^n$, $\lambda_1 \in \mathbb{R}^l$, $\sigma_1 > 0$, $0 \le \varepsilon \ll 1$, $\vartheta \in (0, 1)$, $\eta > 1$. 令 $k := 1$.

步 1 求解子问题. 以 x_{k-1} 为初始点求解无约束子问题的极小点 x_k:

$$\min \psi(x, \lambda_k, \sigma_k) = f(x) - \lambda_k^T h(x) + \frac{\sigma_k}{2}\|h(x)\|^2. \tag{9.27}$$

步 2 检验终止条件. 若 $\|h(x_k)\| \le \varepsilon$, 停算, 输出 x_k 作为原问题的近似极小点; 否则, 转步 3.

步 3　更新罚参数. 若 $\|h(x_k)\| \ge \vartheta\|h(x_{k-1})\|$, 令 $\sigma_{k+1} := \eta\sigma_k$; 否则, $\sigma_{k+1} := \sigma_k$.

步 4　更新乘子向量. 令 $\lambda_{k+1} = \lambda_k - \sigma_k h(x_k)$.

步 5　令 $k := k+1$, 转步 1.

下面我们来看一个简单的例子.

例 27　用乘子法求解约束优化问题

$$\begin{cases} \min & f(x) = x_1^2 - x_2^2, \\ \text{s.t.} & x_2 = -1. \end{cases}$$

解　首先写出所求问题相应于乘子法的增广目标函数:

$$\begin{aligned} \psi(x,\lambda,\sigma) &= x_1^2 - x_2^2 - \lambda(x_2+1) + \frac{\sigma}{2}(x_2+1)^2 \\ &= x_1^2 + \left(\frac{\sigma}{2} - 1\right)x_2^2 + (\sigma-\lambda)x_2 - \lambda - \frac{\sigma}{2}. \end{aligned}$$

令

$$\frac{\partial\psi}{\partial x_1} = 2x_1 = 0, \quad \frac{\partial\psi}{\partial x_2} = (\sigma-2)x_2 - (\lambda-\sigma) = 0.$$

对于 $\sigma > 2$, 解上述关于 x_1 和 x_2 的二元一次方程组得稳定点

$$\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \dfrac{\lambda-\sigma}{\sigma-2} \end{pmatrix}.$$

注意到约束条件 $x_2 = -1$, 可得 $\lambda = 2$. 从而 $\bar{x} = (0,-1)^T = x^*$.

从上面的例子可以发现, 乘子法并不要求罚参数 σ 趋于无穷大, 只要求它大于某个正数即可. 下面我们从理论上来证明这一事实.

引理 16　已知矩阵 $U \in \mathbb{R}^{n\times n}$, $S \in \mathbb{R}^{n\times m}$. 则对任意满足 $S^Tx = 0$ 的非零向量 x 都有 $x^TUx > 0$ 的充要条件是存在常数 $\sigma^* > 0$, 使得对任意的 $\sigma \ge \sigma^*$ 有,

$$x^T(U + \sigma SS^T)x > 0, \quad \forall\, 0 \ne x \in \mathbb{R}^n.$$

证　充分性是显然的. 注意到 $S^Tx=0$, 立即有

$$x^TUx = x^TUx + \sigma x^TSS^Tx = x^T(U+\sigma SS^T)x > 0.$$

必要性. 注意到, 若存在常数 $\sigma^* > 0$, 使得

$$x^T(U+\sigma^* SS^T)x > 0, \quad \forall\, 0 \neq x \in \mathbb{R}^n,$$

则任取 $\sigma \geq \sigma^*$, 恒有

$$x^T(U+\sigma SS^T)x \geq x^T(U+\sigma^* SS^T)x > 0, \quad \forall\, 0 \neq x \in \mathbb{R}^n.$$

以下只需证明上述 σ^* 的存在性. 事实上, 若这样的 σ^* 不存在, 则对任意的正整数 k, 必存在向量 x_k 且 $\|x_k\|=1$, 使得

$$x_k^T(U+kSS^T)x_k \leq 0.$$

因 $\{x_k\}$ 是有界序列, 故必有收敛的子序列 (仍记为它本身), 其极限为 $\hat{x}$, 切 $\|\hat{x}\|=1$. 那么, 对上式取极限, $k\to\infty$, 得

$$\hat{x}^TU\hat{x} + \lim_{k\to\infty} k\|S^Tx_k\|^2 \leq 0.$$

由此, 必有 $\|S^Tx_k\| \to \|S^T\hat{x}\| = 0$ (否则, $k\|S^Tx_k\|^2 \to \infty$), 同时 $\hat{x}^TU\hat{x} \leq 0$, 这与必要性的条件矛盾. 证毕. □

定理 48　设等式约束优化问题 (9.22) 的 KT 点 (x^*, λ^*) 满足二阶充分性条件 (参见定理 37), 则存在一个 $\sigma^* > 0$, 对所有的 $\sigma \geq \sigma^*$, x^* 是增广目标函数 $\psi(x,\lambda^*,\sigma)$ (由 (9.24) 所定义) 的严格局部极小点. 进一步, 若 $h(\bar{x})=0$, 且 $\bar{x}$ 对某个 $\bar{\lambda}$ 是 $\psi(x,\bar{\lambda},\sigma)$ 的局部极小点, 则 $\bar{x}$ 也是问题 (9.22) 的局部极小点.

证　注意到 $\psi(x,\lambda^*,\sigma) = L(x,\lambda^*) + \dfrac{\sigma}{2}\|h(x)\|^2$, 求其梯度和二阶导数得

$$\begin{aligned}
\nabla_x\psi(x,\lambda^*,\sigma) &= \nabla_x L(x,\lambda^*) + \sigma\nabla h(x)\,h(x), \\
\nabla_x^2\psi(x^*,\lambda^*,\sigma) &= \nabla_x^2 L(x^*,\lambda^*) + \sigma\nabla h(x^*)\nabla h(x^*)^T.
\end{aligned}$$

由二阶充分性条件知, 对每个满足 $d^T\nabla h(x^*)=0$ 的向量 $d\neq 0$, 有

$$d^T\nabla_x^2 L(x^*,\lambda^*)d > 0.$$

于是由引理 16 知, 存在 $\sigma^* > 0$ 使得当 $\sigma \geq \sigma^*$ 且 $d \neq 0$ 时, 有

$$d^T\nabla_x^2\psi(x^*,\lambda^*)d = d^T[\nabla_x^2 L(x^*,\lambda^*) + \sigma\nabla h(x^*)\nabla h(x^*)^T]d > 0.$$

再由二阶充分性条件知, $\nabla_x L(x^*,\lambda^*) = 0$, $h(x^*) = 0$. 故而有

$$\nabla_x\psi(x^*,\lambda^*,\sigma) = \nabla_x L(x^*,\lambda^*,\sigma) = 0,$$

这表明增广目标函数 $\psi(x,\lambda^*,\sigma)\,(\sigma \geq \sigma^*)$ 在 x^* 处满足二阶充分性条件, 故由定理 ?? 可知, x^* 是 $\psi(x,\lambda^*,\sigma)$ 的严格局部极小点. 这就证明了定理的第一个结论.

下面我们来证明第二个结论. 若 $h(\bar{x}) = 0$, 且 $\bar{x}$ 对某个 $\bar{\lambda}$ 是 $\psi(x,\bar{\lambda},\sigma)$ 的局部极小点. 则对任意与 $\bar{x}$ 充分靠近的 $\hat{x}$ (即 $h(\hat{x}) = 0$), 有

$$\psi(\bar{x},\bar{\lambda},\sigma) \leq \psi(\hat{x},\bar{\lambda},\sigma).$$

因 $h(\bar{x}) = h(\hat{x}) = 0$, 故有

$$\psi(\bar{x},\bar{\lambda},\sigma) = f(\bar{x}), \quad \psi(\hat{x},\bar{\lambda},\sigma) = f(\hat{x}).$$

上式表明, 对于 $\bar{x}$ 的某个邻域中的任意可行点 $\hat{x}$, 均有 $f(\bar{x}) \leq f(\hat{x})$, 即 $\bar{x}$ 是问题 (9.22) 的局部极小点. 证毕. □

2. 一般约束问题的乘子法

下面我们考虑同时带有等式和不等式约束的优化问题的乘子法:

$$\begin{cases} \min\ f(x), \\ \text{s.t.}\ \ h_i(x) = 0,\ i = 1,\cdots,l, \\ \qquad g_i(x) \geq 0,\ i = 1,\cdots,m. \end{cases} \tag{9.28}$$

其基本思想是把解等式约束优化问题的乘子法推广到不等式约束优化问题, 即先引进辅助变量把不等式约束化为等式约束, 然后再利用最优性条件消去辅助变量. 为叙述的方便计, 我们先考虑如下只带有不等式约束的最优化问题

$$\begin{cases} \min\ f(x), \\ \text{s.t.}\ \ g_i(x) \geq 0,\ i = 1,\cdots,m. \end{cases}$$

引进辅助变量 $y_i\,(i=1,\cdots,m)$, 可以将上面的优化问题化为等价的等式约束优化问题:

$$\begin{cases} \min\ f(x), \\ \text{s.t.}\ \ g_i(x)-y_i^2=0,\ i=1,\cdots,m. \end{cases}$$

利用算法 21 求解, 此时增广拉格朗日函数为

$$\tilde{\psi}(x,y,\lambda,\sigma)=f(x)-\sum_{i=1}^{m}\lambda_i[g_i(x)-y_i^2]+\frac{\sigma}{2}\sum_{i=1}^{m}[g_i(x)-y_i^2]^2.$$

为了消去辅助变量 y, 可考虑 $\tilde{\psi}$ 关于变量 y 的极小化. 由一阶必要条件, 令 $\nabla_y\tilde{\psi}(x,y,\lambda,\sigma)=0$ 可得

$$2y_i\lambda_i-2\sigma y_i[g_i(x)-y_i^2]=0,\ \ i=1,\cdots,m,$$

即

$$y_i[\sigma y_i^2-(\sigma g_i(x)-\lambda_i)]=0,\ \ i=1,\cdots,m.$$

故当 $\sigma g_i(x)-\lambda_i>0$ 时, 有

$$y_i^2=\frac{1}{\sigma}\big[\sigma g_i(x)-\lambda_i\big]=g_i(x)-\frac{1}{\sigma}\lambda_i.$$

否则, 由 $\sigma y_i^2+[\lambda_i-\sigma g_i(x)]\ge 0$ 可推得 $y_i=0$. 综合起来, 有

$$y_i^2=\begin{cases} \dfrac{1}{\sigma}\big[\sigma g_i(x)-\lambda_i\big], & \sigma g_i(x)-\lambda_i>0, \\ 0, & \sigma g_i(x)-\lambda_i\le 0, \end{cases}\qquad i=1,\cdots,m.$$

即有

$$g_i(x)-y_i^2=\begin{cases} \dfrac{\lambda_i}{\sigma}, & \sigma g_i(x)-\lambda_i>0, \\ g_i(x), & \sigma g_i(x)-\lambda_i\le 0, \end{cases}\qquad i=1,\cdots,m. \tag{9.29}$$

因此, 当 $\sigma g_i(x)-\lambda_i\le 0$ 时, 我们有

$$\begin{aligned} &-\lambda_i[g_i(x)-y_i^2]+\frac{\sigma}{2}[g_i(x)-y_i^2]^2 \\ &=-\lambda_i g_i(x)+\frac{\sigma}{2}[g_i(x)]^2 \\ &=\frac{1}{2\sigma}\big[(\sigma g_i(x)-\lambda_i)^2-\lambda_i^2\big]. \end{aligned}$$

而当 $\sigma g_i(x)-\lambda_i>0$ 时, 有

$$\begin{aligned}&-\lambda_i[g_i(x)-y_i^2]+\frac{\sigma}{2}[g_i(x)-y_i^2]^2\\&=-\frac{1}{\sigma}\lambda_i^2+\frac{1}{2\sigma}\lambda_i^2=-\frac{1}{2\sigma}\lambda_i^2.\end{aligned}$$

综合上述两种情形, 有

$$-\lambda_i[g_i(x)-y_i^2]+\frac{\sigma}{2}[g_i(x)-y_i^2]^2=\frac{1}{2\sigma}\Big([\min\{0,\sigma g_i(x)-\lambda_i\}]^2-\lambda_i^2\Big).$$

将其代回到 $\tilde{\psi}(x,y,\lambda,\sigma)$ 中去得

$$\begin{aligned}\tilde{\phi}(x,\lambda,\sigma)&=\min_y\tilde{\psi}(x,y,\lambda,\sigma)\\&=f(x)+\frac{1}{2\sigma}\sum_{i=1}^m\Big([\min\{0,\sigma g_i(x)-\lambda_i\}]^2-\lambda_i^2\Big).\end{aligned}\tag{9.30}$$

于是, 将 (9.29) 代入乘子迭代公式 $(\lambda_{k+1})_i=(\lambda_k)_i-\sigma[g_i(x_k)-(y_k)_i^2]$ 得

$$(\lambda_{k+1})_i=\begin{cases}0, & \sigma g_i(x_k)-(\lambda_k)_i>0,\\(\lambda_k)_i-g_i(x_k), & \sigma g_i(x_k)-(\lambda_k)_i\le 0,\end{cases}$$

即

$$(\lambda_{k+1})_i=\max\{0,(\lambda_k)_i-g_i(x_k)\}\ge 0,\quad i=1,\cdots,m.\tag{9.31}$$

同样, 在终止准则

$$\Big(\sum_{i=1}^m[g_i(x_k)-(y_k)_i^2]\Big)^{1/2}\le\varepsilon$$

中代入 (9.29) 得

$$\left(\sum_{i=1}^m\left[\min\left\{g_i(x_k),\frac{(\lambda_k)_i}{\sigma}\right\}\right]\right)^{1/2}\le\varepsilon.\tag{9.32}$$

现在我们回到一般约束优化问题 (9.28), 我们来构造求解 (9.28) 的乘子法. 此时, 增广拉格朗日函数为

$$\begin{aligned}\psi(x,\mu,\lambda,\sigma)&=f(x)-\sum_{i=1}^l\mu_ih_i(x)+\frac{\sigma}{2}\sum_{i=1}^lh_i^2(x)\\&\quad+\frac{1}{2\sigma}\sum_{i=1}^m\big([\min\{0,\sigma g_i(x)-\lambda_i\}]^2-\lambda_i^2\big).\end{aligned}\tag{9.33}$$

乘子迭代的公式为

$$\begin{aligned}(\mu_{k+1})_i &= (\mu_k)_i - \sigma h_i(x_k), \quad i=1,\cdots,l, \\ (\lambda_{k+1})_i &= \max\{0, (\lambda_k)_i - g_i(x_k)\}, \quad i=1,\cdots,m.\end{aligned}$$

令

$$\beta_k = \left(\sum_{i=1}^{l} h_i^2(x_k) + \sum_{i=1}^{m} \left[\min\left\{ g_i(x_k), \frac{(\lambda_k)_i}{\sigma} \right\} \right]^2 \right)^{1/2}, \tag{9.34}$$

则终止准则为

$$\beta_k \le \varepsilon.$$

下面给出求解一般约束优化问题 (9.28) 乘子法的详细步骤. 由于这一算法是由 Rockfellar 在 PH 算法的基础上提出的, 因此简称为 PHR 算法.

算法 24 PHR 算法

步 0　选取初始值. 给定 $x_0 \in \mathbb{R}^n$, $\mu_1 \in \mathbb{R}^l$, $\lambda_1 \in \mathbb{R}^m$, $\sigma_1 > 0$, $0 \le \varepsilon \ll 1$, $\vartheta \in (0,1)$, $\eta > 1$. 令 $k := 1$.

步 1　求解子问题. 以 x_{k-1} 为初始点求解无约束子问题

$$\min \psi(x, \mu_k, \lambda_k, \sigma_k),$$

得极小点 x_k, 其中 $\psi(x, \mu_k, \lambda_k, \sigma_k)$ 由 (9.33) 定义.

步 2　检验终止条件. 若 $\beta_k \le \varepsilon$, 其中 β_k 由 (9.34) 定义, 则停止迭代, 输出 x_k 作为原问题的近似极小点; 否则, 转步 3.

步 3　更新罚参数. 若 $\beta_k \ge \vartheta \beta_{k-1}$, 令 $\sigma_{k+1} := \eta \sigma_k$; 否则, $\sigma_{k+1} := \sigma_k$.

步 4　更新乘子向量. 计算

$$\begin{aligned}(\mu_{k+1})_i &= (\mu_k)_i - \sigma h_i(x_k), \quad i=1,\cdots,l, \\ (\lambda_{k+1})_i &= \max\{0, (\lambda_k)_i - g_i(x_k)\}, \quad i=1,\cdots,m.\end{aligned}$$

步 5　令 $k := k+1$, 转步 1.

9.4　乘子法的 Matlab 实现

本小节我们给出算法 24 (PHR 乘子法) 的 Matlab 通用程序, 然后利用该程序求解一个简单的约束优化问题.

程序 15（PHR 算法程序）

```
function [x,mu,lambda,output]=multphr(fun,hf,gf,dfun,dhf,dgf,x0)
%功能: 用乘子法解一般约束问题:  min f(x), s.t. h(x)=0, g(x)>=0
%输入:  x0是初始点, fun, dfun分别是目标函数及其梯度;
%   hf, dhf分别是等式约束（向量）函数及其Jacobi矩阵的转置;
%   gf, dgf分别是不等式约束（向量）函数及其Jacobi矩阵的转置;
%输出:  x是近似最优点, mu, lambda分别是相应于等式约束和不
%   等式约束的乘子向量; output是结构变量, 输出近似极小值f, 迭
%   代次数, 内迭代次数等
maxk=500;   %最大迭代次数
sigma=2.0;  %罚因子
eta=2.0;  theta=0.8;  %PHR算法中的实参数
k=0; ink=0;  %k, ink分别是外迭代和内迭代次数
epsilon=1e-5;  %终止误差值
x=x0;  he=feval(hf,x); gi=feval(gf,x);
n=length(x); l=length(he); m=length(gi);
%选取乘子向量的初始值
mu=0.1*ones(l,1);  lambda=0.1*ones(m,1);
btak=10;  btaold=10;  %用来检验终止条件的两个值
while(btak>epsilon & k<maxk)
    %调用BFGS算法程序求解无约束子问题
    [x,ival,ik]=bfgs('mpsi','dmpsi',x0,fun,hf,gf,dfun,dhf,dgf,mu,lambda,sigma);
    ink=ink+ik;
    he=feval(hf,x); gi=feval(gf,x);
    btak=0.0;
    for(i=1:l), btak=btak+he(i)^2;   end
    for(i=1:m)
        temp=min(gi(i),lambda(i)/sigma);
        btak=btak+temp^2;
    end
    btak=sqrt(btak);
    if btak>epsilon
        if(k>=2 & btak > theta*btaold)
            sigma=eta*sigma;
        end
        %更新乘子向量
        for(i=1:l),  mu(i)=mu(i)-sigma*he(i); end
```

```
        for(i=1:m)
            lambda(i)=max(0.0,lambda(i)-sigma*gi(i));
        end
    end
    k=k+1;
    btaold=btak;
    x0=x;
end
f=feval(fun,x);
output.fval=f;
output.iter=k;
output.inner_iter=ink;
output.bta=btak;
%%%%%%%%%%%%%%%%%%  增广拉格朗日函数 %%%%%%%%%%%%%%%%%%%%%
function psi=mpsi(x,fun,hf,gf,dfun,dhf,dgf,mu,lambda,sigma)
f=feval(fun,x);  he=feval(hf,x);  gi=feval(gf,x);
l=length(he); m=length(gi);
psi=f;  s1=0.0;
for(i=1:l)
    psi=psi-he(i)*mu(i);
    s1=s1+he(i)^2;
end
psi=psi+0.5*sigma*s1;
s2=0.0;
for(i=1:m)
    s3=max(0.0, lambda(i) - sigma*gi(i));
    s2=s2+s3^2-lambda(i)^2;
end
psi=psi+s2/(2.0*sigma);
%%%%%%%%%%%%%%%%%%  增广拉格朗日函数的梯度%%%%%%%%%%%%%%%%%%%%
function dpsi=dmpsi(x,fun,hf,gf,dfun,dhf,dgf,mu,lambda,sigma)
dpsi=feval(dfun,x);
he=feval(hf,x);  gi=feval(gf,x);
dhe=feval(dhf,x);  dgi=feval(dgf,x);
l=length(he); m=length(gi);
for(i=1:l)
    dpsi=dpsi+(sigma*he(i)-mu(i))*dhe(:,i);
```

```
end
for(i=1:m)
    dpsi=dpsi+(sigma*gi(i)-lambda(i))*dgi(:,i);
end
```

例 28 利用程序 15 求解约束优化问题

$$\begin{aligned} \min\ & f(x)=(x_1-2)^2+(x_2-1)^2, \\ \text{s.t.}\ & x_1-2x_2+1=0, \\ & 0.25x_1^2+x_2^2\le 1. \end{aligned}$$

取初始点为 $x_0=(3,3)^T$，该问题有精确解 $x^*=\left(\frac{1}{2}(\sqrt{7}-1),\frac{1}{4}(\sqrt{7}+1)\right)^T$ $\approx(0.82288,0.91144)^T$，$f(x^*)=\frac{1}{4}\left(\sqrt{7}-5\right)^2+\frac{1}{16}\left(\sqrt{7}-3\right)^2\approx 1.39346.$

解 先编制目标函数文件 f1.m

```
function f=f1(x)
f=(x(1)-2.0)^2+(x(2)-1.0)^2;
```

等式约束函数文件 h1.m

```
function he=h1(x)
he=x(1)-2.0*x(2)+1.0;
```

不等式约束函数文件 g1.m

```
function gi=g1(x)
gi=-0.25*x(1)^2-x(2)^2+1;
```

目标函数的梯度文件 df1.m

```
function g=df1(x)
g=[2.0*(x(1)-2.0), 2.0*(x(2)-1.0)]';
```

等式约束（向量）函数的Jacobi矩阵（转置）文件 dh1.m

```
function dhe=dh1(x)
dhe=[1.0, -2.0]';
```

不等式约束（向量）函数的Jacobi矩阵（转置）文件 dg1.m

```
function dgi=dg1(x)
dgi=[-0.5*x(1), -2.0*x(2)]';
```

然后在 Matlab 命令窗口输入如下命令:

```
x0=[3,3]';
[x,mu,lambda,output]=multphr('f1','h1','g1','df1','dh1','dg1',x0);
```

得到如下输出:

```
x =
    0.8229
    0.9114
mu =
  -1.5945
lambda =
   1.8465
output =
          fval: 1.3934
          iter: 23
    inner_iter: 82
           bta: 9.5419e-006
```

习　题　9

1. 用外罚函数法求解下列约束优化问题:

(1) $\min\ f(x) = -x_1 - x_2$, s.t. $x_1^2 + x_2^2 = 1$,

(2) $\min\ f(x) = x_1^2 + x_2^2$, s.t. $x_1 + x_2 = 1$,

(3) $\min\ f(x) = x_1^2 + x_2^2$, s.t. $2x_1 + x_2 - 2 \le 0$, $x_2 \ge 1$,

(4) $\min\ f(x) = -x_1 x_2$, s.t. $-x_1 - x_2^2 + 1 \le 0$, $x_1 + x_2 \ge 0$.

2. 用内点法求解下列约束优化问题

(1) $\min\ f(x) = x_1 + x_2$, s.t. $-x_1^2 + x_2 \le 0$, $x_1 \ge 0$,

(2) $\min\ f(x) = x_1^2 + x_2$, s.t. $-2x_1 - x_2 + 2 \ge 0$, $x_2 - 1 \ge 0$,

(3) $\min \quad f(x) = 2x_1 + 3x_2,$ $\quad$ s.t. $\quad 1 - 2x_1^2 - x_2^2 \geq 0,$

(4) $\min \quad f(x) = (x+1)^2,$ $\quad$ s.t. $\quad x \geq 0.$

3. 用乘子法求解下列问题

(1) $\min \quad f(x) = x_1^2 + x_2^2,$ $\quad$ s.t. $\quad x_1 \geq 0,$

(2) $\min \quad f(x) = \frac{1}{2}x_1^2 + \frac{1}{6}x_2^2$ $\quad$ s.t. $\quad x_1 + x_2 - 1 = 0,$

(3) $\min \quad f(x) = x_1 + \frac{1}{3}(x_2+1)^2,$ $\quad$ s.t. $\quad x_1 \geq 0,$ $\quad x_2 \geq 1,$

(4) $\min \quad f(x) = (x_1-2)^2 + (x_2-3)^2,$ $\quad$ s.t. $\quad (x_1-2)^2 - x_2 \geq 0,$ $\quad 2x_1 - x_2 - 1 = 0.$

4. 考虑下列问题

$$\begin{aligned} \min \quad & f(x) = x_1 x_2, \\ \text{s.t.} \quad & g(x) = -2x_1 + x_2 + 3 \geq 0. \end{aligned}$$

(1) 用二阶最优性条件证明点 $\bar{x} = \left(\frac{3}{4}, -\frac{3}{2}\right)^T$ 是局部极小点, 并说明它是否为全局极小点.

(2) 定义障碍函数为 $G(x,r) = x_1 x_2 - r\ln g(x)$, 试用内点法求解此问题, 并说明内点法产生的序列趋向于 $\bar{x}$.

5. 考虑下列非线性约束优化问题:

$$\begin{aligned} \min \quad & x_1^3 + x_2^3 \\ \text{s.t.} \quad & x_1 + x_2 = 1. \end{aligned}$$

(1) 求问题的最优解.

(2) 定义罚函数 $P(x,\sigma) = x_1^3 + x_2^3 + \sigma(x_1 + x_2 - 1)^2$, 试讨论能否通过求解无约束问题 $\min P(x,\sigma)$ 来获得原问题的最优解? 说明理由.

6. 利用乘子法的 Matlab 程序求解下列约束优化问题:

(1)
$$\begin{aligned} \min \quad & f(x) = (x_1-2)^2 + (x_2-1)^2, \\ \text{s.t.} \quad & x_1 - 2x_2 + 1 = 0, \\ & -0.25x_1^2 - x_2^2 + 1 \geq 0. \end{aligned}$$

初始点取为 $(2,2)^T$, 极小点为 $\big(0.5(\sqrt{7}-1), 0.25(\sqrt{7}+1)\big)^T$.

(2)
$$\begin{aligned} \min \quad & f(x) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3, \\ \text{s.t.} \quad & 8x_1 + 14x_2 + 7x_3 - 56 = 0, \\ & x_1^2 + x_2^2 + x_3^2 - 25 = 0 \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0. \end{aligned}$$

初始点取为 $(2,2,2)^T$, 极小点为 $(3.512118,\ 0.216988,\ 3.552174)^T$.

第十章　可行方向法

本章介绍的可行方向法是一类直接处理约束优化问题的方法, 其基本思想是要求每一步迭代产生的搜索方向不仅对目标函数是下降方向, 而且对约束函数来说是可行方向, 即迭代点总是满足所有的约束条件. 各种不同的可行方向法的主要区别在于选取可行方向 d_k 的策略不同, 我们主要介绍 Zoutendijk 可行方向法、投影梯度法和简约梯度法三种类可行方向法.

10.1　Zoutendijk 可行方向法

Zoutendijk 可行方向法是用一个线性规划来确定搜索方向 — 下降可行方向的方法, 它最早是由 Zoutendijk 于 1960 年提出来的. 我们分线性约束和非线性约束两种情形来讨论其算法原理.

10.1.1　线性约束下的可行方向法

1. 基本原理

考虑下面的非线性优化问题

$$\begin{cases} \min\ f(x), \\ \text{s.t.}\ Ax \geq b, \\ \qquad Ex = e, \end{cases} \tag{10.1}$$

其中 $f(x)$ 连续可微, A 是 $m \times n$ 矩阵, E 是 $l \times n$ 矩阵, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $e \in \mathbb{R}^l$. 即 (10.1) 中有 m 个线性不等式约束和 l 个线性等式约束.

下面的引理指出了问题 (10.1) 的下降可行方向 d 应满足的条件.

引理 17 设 $\bar{x}$ 是问题 (10.1) 的一个可行点, 且在 $\bar{x}$ 处有 $A_1\bar{x}=b_1$, $A_2\bar{x}>b_2$, 其中

$$A=\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \quad b=\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

则 $d\in\mathbb{R}^n$ 是点 $\bar{x}$ 处的下降可行方向的充要条件是

$$A_1 d\geq 0,\ \ Ed=0,\ \ \nabla f(\bar{x})^T d<0.$$

证 不难发现 d 是 $f(x)$ 在 $\bar{x}$ 处的下降方向的充要条件是 $\nabla f(\bar{x})^T d<0$. 另外, 注意到条件 $A_1\bar{x}=b_1$ 表明约束条件 $A_1x\geq b_1$ 是点 $\bar{x}$ 出的有效约束, 而条件 $A_2\bar{x}>b_2$ 表明约束条件 $A_2x\geq b_2$ 是点 $\bar{x}$ 处的非有效约束. 因此, 在可行点 $\bar{x}$ 处将约束矩阵 A 分裂为相应的 A_1 和 A_2 两部分.

充分性. 设 $A_1d\geq 0$, $Ed=0$. 因 $\bar{x}$ 是可行点, 且 $A_1\bar{x}=b_1$, $E\bar{x}=e$. 故对任意的 $\alpha>0$, 都有

$$\begin{aligned} A_1(\bar{x}+\alpha d) &= A_1\bar{x}+\alpha(A_1d)\geq A_1\bar{x}=b_1, \\ E(\bar{x}+\alpha d) &= E\bar{x}+\alpha(Ed)=E\bar{x}=e. \end{aligned}$$

又由 $A_2\bar{x}>b_2$, 故必存在一个 $\bar{\alpha}>0$, 使得对于任意的 $\alpha\in(0,\bar{\alpha}]$, 都有

$$A_2(\bar{x}+\alpha d)=A_2\bar{x}+\alpha A_2 d\geq b_2.$$

以上三式表明, 存在 $\bar{\alpha}$, 使得对于任意的 $\alpha\in(0,\bar{\alpha}]$, 有

$$A(\bar{x}+\alpha d)\geq b,\ \ E(\bar{x}+\alpha d)=e,$$

即 $\bar{x}+\alpha d$ 是可行点, 从而 d 是 $\bar{x}$ 处的可行方向.

必要性. 设 $\bar{x}$ 是可行点, d 是 $\bar{x}$ 处的一个可行方向. 由可行方向的定义, 存在 $\bar{\alpha}$, 使得对于任意的 $\alpha\in(0,\bar{\alpha}]$, 有

$$A(\bar{x}+\alpha d)\geq b,\ \ E(\bar{x}+\alpha d)=e,$$

或

$$A_1(\bar{x}+\alpha d)\geq b_1,\ \ A_2(\bar{x}+\alpha d)\geq b_2,\ \ E(\bar{x}+\alpha d)=e.$$

于是由

$$A_1(\bar{x}+\alpha d)=A_1\bar{x}+\alpha(A_1d)\geq b_1,\ \ A_1\bar{x}=b_1,\ \ \alpha>0$$

可推出 $A_1 d \geq 0$. 又由

$$E(\bar{x} + \alpha d) = E\bar{x} + \alpha(A_1 d) \geq e, \quad E\bar{x} = e, \quad \alpha > 0$$

可推出 $Ed = 0$. 证毕. □

上面的引理启发我们, 要寻找问题 (10.1) 的可行点 $\bar{x}$ 处的一个下降可行方向 d, 可以通过求解下述线性规划问题得到:

$$\begin{cases} \min\ \nabla f(\bar{x})^T d, \\ \text{s.t.}\ \ A_1 d \geq 0, \\ \qquad\ \ Ed = 0, \\ \qquad\ \ -1 \leq d_i \leq 1,\ i = 1, \cdots, n, \end{cases} \tag{10.2}$$

其中 $d = (d_1, \cdots, d_n)^T$. 增加约束条件 $-1 \leq d_i \leq 1$, $i = 1, \cdots, n$ 是为了防止 $\|d\| \to \infty$.

注意到, $d = 0$ 显然是子问题 (10.2) 的一个可行解, 故目标函数 $\nabla f(\bar{x})^T d$ 的最优值必然小于或等于 0. 若目标函数的最优值 $\bar{z} = \nabla f(\bar{x})^T \bar{d} < 0$, 则由引理 17 可知, $\bar{d}$ 即为 $\bar{x}$ 处的下降可行方向. 否则, 若标函数的最优值 $\bar{z} = \nabla f(\bar{x})^T \bar{d} = 0$, 则可以证明 $\bar{x}$ 是问题 (10.1) 的 KT 点.

定理 49 设 $\bar{x}$ 是问题 (10.1) 的一个可行点, 且在 $\bar{x}$ 处有 $A_1 \bar{x} = b_1$, $A_2 \bar{x} > b_2$, 其中

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

则 $\bar{x}$ 是问题 (10.1) 的 KT 点的充要条件是子问题 (10.2) 的最优值为 0.

由于上述定理的证明需要用到 Farkas 引理 (引理 11), 为了使用方便, 我们给出 Farkas 引理的一个等价描述方式.

引理 18 (Farkas 引理) 设 A 为 $m \times n$ 矩阵, c 为 n 维向量. 则 $A^T y = c$, $y \geq 0$ 有解的充分必要条件是 $Ax \geq 0$, $c^T x > 0$ 无解, 其中 x, y 分别是为 n, m 维向量.

定理 49 的证明: 注意到, $\bar{x}$ 是 KT 点充要条件是, 存在 $\lambda \geq 0$ 和 μ, 使得

$$\nabla f(\bar{x}) - A_1^T \lambda - E^T \mu = 0. \tag{10.3}$$

令 $\mu = \nu_1 - \nu_2\,(\nu_1, \nu_2 \geq 0)$, 把 (10.3) 写成

$$\big(-A_1^T, -E^T, E^T\big)\begin{pmatrix}\lambda\\ \nu_1\\ \nu_2\end{pmatrix} = -\nabla f(\bar{x}), \quad \begin{pmatrix}\lambda\\ \nu_1\\ \nu_2\end{pmatrix} \geq 0. \tag{10.4}$$

根据 Farkas 引理, (10.4) 有解的充要条件是

$$\begin{pmatrix}-A_1\\ -E\\ E\end{pmatrix} d \leq 0, \quad -\nabla f(\bar{x})^T d > 0 \tag{10.5}$$

无解, 即

$$A_1 d \geq 0, \quad Ed = 0, \quad \nabla f(\bar{x})^T d < 0$$

无解. 故 $\bar{x}$ 是问题 (10.1) 的 KT 点的充要条件是子问题 (10.2) 的最优值为 0. □

由上述定理可知, 求解子问题 (10.2) 的结果, 或者得到下降可行方向, 或者得到原问题的一个 KT 点.

2. 计算步骤

下面讨论可行方向法的具体计算步骤. 首先分析如何确定搜索步长 α_k. 设问题的可行域为 $\mathcal{F}$. 第 k 次迭代的出发点 $x_k \in \mathcal{F}$ 是可行点, d_k 是其下降可行方向, 则后继点 x_{k+1} 为

$$x_{k+1} = x_k + \alpha_k d_k. \tag{10.6}$$

为了使得 $x_{k+1} \in \mathcal{F}$, 且使 $f(x_{k+1})$ 的值尽可能小, 可以通过求解下面的一维搜索问题来解决:

$$\begin{aligned}&\min_{0\leq\alpha\leq\bar{\alpha}} f(x_k + \alpha d_k),\\ &\bar{\alpha} = \max\{\alpha | x_k + \alpha d_k \in \mathcal{F}\}.\end{aligned} \tag{10.7}$$

在求解 (10.7) 式时, 考虑到线性约束情形时的 (10.1) 式, 先求解

$$\begin{cases}\min & f(x_k + \alpha d_k),\\ \text{s.t.} & A(x_k + \alpha d_k) \geq b,\\ & E(x_k + \alpha d_k) = e.\end{cases} \tag{10.8}$$

而 (10.8) 可作进一步的简化: 因为 d_k 是可行方向, 有 $Ed_k = 0$; 而 x_k 是可行点, 有 $Ex_k = e$. 因此 (10.8) 中的等式约束条件自然成立, 可不必再考虑它. 此外, 在 x_k 处, 将不等式约束分为有效约束和非有效约束, 设

$$A_1 x_k = b_1, \quad A_2 x_k > b_2, \tag{10.9}$$

其中

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

则 (10.8) 中的不等式约束条件可分裂成:

$$A_1 x_k + \alpha A_1 d_k \geq b_1, \tag{10.10}$$

$$A_2 x_k + \alpha A_2 d_k \geq b_2. \tag{10.11}$$

又因 d_k 是可行方向, 由引理 17 知 $A_1 d_k \geq 0$. 注意到 $A_1 x_k = b_1$ 及 $\alpha \geq 0$, 因此条件 (10.10) 也自然成立. 于是 (10.8) 中的约束条件只剩下 (10.11), 故 (10.8) 可简化为:

$$\begin{cases} \min & f(x_k + \alpha d_k), \\ \text{s.t.} & A_2(x_k + \alpha d_k) > b_2, \\ & \alpha \geq 0. \end{cases} \tag{10.12}$$

以下讨论 (10.12) 中求 α 上限的公式. 将 (10.12) 中的第一个约束条件改写成:

$$\alpha A_2 d_k \geq b_2 - A_2 x_k.$$

若记

$$\bar{b} = b_2 - A_2 x_k, \quad \bar{d} = A_2 d_k, \tag{10.13}$$

则有

$$\alpha \bar{d} \geq \bar{b}, \quad \alpha \geq 0.$$

注意到 (10.9) 式, 我们有 $\bar{b} < 0$. 由此可得 α 的上界计算公式:

$$\bar{\alpha} = \begin{cases} \min\left\{ \dfrac{\bar{b}_i}{\bar{d}_i} = \dfrac{(b_2 - A_2 x_k)_i}{(A_2 d_k)_i} \;\middle|\; \bar{d}_i < 0 \right\}, & \bar{d} \ngeq 0, \\ +\infty, & \bar{d} \geq 0, \end{cases} \tag{10.14}$$

其中 $\bar{b}_i$, $\bar{d}_i$ 分别是向量 $\bar{b}$, $\bar{d}$ 的第 i 个分量. 因此, 求解 (10.12) 等价于求解

$$\begin{cases} \min \quad f(x_k+\alpha d_k), \\ \text{s.t.} \quad 0\le \alpha \le \bar{\alpha}, \end{cases} \tag{10.15}$$

其中 $\bar{\alpha}$ 由 (10.14) 式计算.

至此, 我们可以写出求解问题 (10.1) 的可行方向法的详细计算步骤:

算法 25 (线性约束的可行方向法)

步 0 给定初始可行点 x_0, 终止误差 $0<\varepsilon_1\ll 1$, $0<\varepsilon_2\ll 1$. 令 $k:=0$.

步 1 在 x_k 处, 将不等式约束分为有效约束和非有效约束:

$$A_1x_k=b_1, \quad A_2x_k>b_2,$$

其中

$$A=\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \quad b=\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

步 2 若 x_k 是可行域的一个内点 (此时问题 (10.1) 中没有等式约束, 即 $E=0$ 且 $A_1=0$), 且 $\|\nabla f(x^k)\|<\varepsilon_1$, 停算, 得到近似极小点 x_k; 否则, 若 x_k 是可行域的一个内点但 $\|\nabla f(x^k)\|\ge\varepsilon_1$,则取搜索方向 $d_k=-\nabla f(x_k)$, 转步 5(即用目标函数的负梯度方向作为搜索方向再求步长, 此时类似于无约束优化问题). 若 x_k 不是可行域的一个内点, 则转步 3.

步 3 求解线性规划问题

$$\begin{cases} \min\ z=\nabla f(x_k)^T d, \\ \text{s.t.}\ \ A_1 d\ge 0, \\ \qquad Ed=0, \\ \qquad -1\le d_i\le 1,\ i=1,\cdots,n, \end{cases} \tag{10.16}$$

其中 $d=(d_1,\cdots,d_n)^T$. 设求得最优解和最优值分别为 d_k 和 z_k.

步 4 若 $|z_k|<\varepsilon_2$, 停算, 输出 x_k 作为近似极小点. 否则, 以 d_k 作为搜索方向, 转步 5.

步 5　首先由 (10.13) 和 (10.14) 计算 $\bar{\alpha}$, 然后作一维线搜索:

$$\begin{cases} \min \quad f(x_k + \alpha d_k), \\ \text{s.t.} \quad 0 \leq \alpha \leq \bar{\alpha}, \end{cases}$$

求得最优解 α_k.

步 6　置 $x_{k+1} := x_k + \alpha_k d_k$, $k := k+1$, 转步 1.

10.1.2　非线性约束下的可行方向法

1. 基本原理

考虑下面带有非线性不等式约束的优化问题:

$$\begin{cases} \min \quad f(x), \\ \text{s.t.} \quad g_i(x) \geq 0, \ i = 1, \cdots, m, \end{cases} \tag{10.17}$$

其中 $x \in \mathbb{R}^n$, $f(x)$, $g_i(x)\,(i = 1, \cdots, m)$ 都是连续可微的函数.

下面的引理指出了问题 (10.17)　的一个下降可行方向　所应满足的条件.

引理 19　设 $\bar{x}$ 是问题 (10.17) 的一个可行点, 指标集 $I(\bar{x}) = \{i \,|\, g_i(\bar{x}) = 0\}$, $f(x)$, $g_i(x)\,(i \in I(\bar{x}))$ 在 $\bar{x}$ 处可微, $g_i(x)\,(i \notin I(\bar{x}))$ 在 $\bar{x}$ 处连续. 若

$$\nabla f(\bar{x})^T d < 0, \quad \nabla g_i(\bar{x})^T d > 0 \ (i \in I(\bar{x})),$$

那么 d 是问题 (10.17) 在 $\bar{x}$ 处的下降可行方向.

证　由引理 13 中的下降可行方向的代数条件 (8.10) 可知, d 必是问题 (10.17) 在 $\bar{x}$ 处的一个下降可行方向. 证毕. □

由上述引理可知, 问题 (10.17) 在可行点 $\bar{x}$ 处的下降可行方向 d 应满足:

$$\begin{cases} \nabla f(\bar{x})^T d < 0, \\ \nabla g_i(x)^T d > 0, \ i \in I(\bar{x}). \end{cases} \tag{10.18}$$

而在 (10.18) 中引进辅助变量 z 后, 等价于下面的线性不等式组求 d 和 z:

$$\begin{cases} \nabla f(\bar{x})^T d \leq z, \\ -\nabla g_i(x)^T d \leq z,\ i \in I(\bar{x}), \\ z \leq 0. \end{cases} \tag{10.19}$$

注意到, 满足 (10.19) 的下降可行方向 d 及数 z 一般有很多个, 我们自然希望求出能使目标函数下降最多的方向 d. 故而可将 (10.19) 转化为以 z 为目标函数的线性规划问题

$$\begin{cases} \min\ z, \\ \text{s.t.}\ \nabla f(\bar{x})^T d \leq z, \\ \qquad -\nabla g_i(x)^T d \leq z,\ i \in I(\bar{x}), \\ \qquad -1 \leq d_i \leq 1,\ i = 1, \cdots, n, \end{cases} \tag{10.20}$$

其中 $d = (d_1, \cdots, d_n)^T$.

设 (10.20) 的最优解为 $\bar{d}$, 最优值为 $\bar{z}$. 那么, 若 $\bar{z} < 0$, 则 $\bar{d}$ 是问题 (10.17) 在 $\bar{x}$ 处的下降可行方向; 否则, 若 $\bar{z} = 0$, 则下面的定理将证明: 相应的 $\bar{x}$ 必为问题 (10.17) 的 Fritz John 点.

定理 50 设 $\bar{x}$ 是问题 (10.17) 的可行点, $I(\bar{x}) = \{i \,|\, g_i(\bar{x}) = 0\}$. 则 $\bar{x}$ 是问题 (10.17) 的 Fritz John 点的充要条件是子问题 (10.20) 的最优值为 0.

证 对于子问题 (10.20), 其最优值为 0 的充要条件是不等式组

$$\begin{cases} \nabla f(\bar{x})^T d < 0, \\ \nabla g_i(\bar{x})^T d > 0, \quad i \in I(\bar{x}), \end{cases}$$

即

$$\begin{cases} \nabla f(\bar{x})^T d < 0, \\ -\nabla g_i(\bar{x})^T d < 0, \quad i \in I(\bar{x}), \end{cases} \tag{10.21}$$

无解. 根据 Gordan 引理 (引理 12), 不等式组 (10.21) 无解的充要条件是存在不全为 0 的数 $\lambda_0 \geq 0$ 和 $\lambda_i\, (i \in I(\bar{x})$, 使得

$$\lambda_0 \nabla f(\bar{x}) - \sum_{i \in I(\bar{x})} \lambda_i \nabla g_i(\bar{x}) = 0,$$

即 $\bar{x}$ 是问题 (10.17) 的 Fritz John 点. 证毕. □

2. 计算步骤

与线性约束情形一样, 为了确定搜索步长 α_k, 仍然需要求解一个一维搜索问题:

$$\begin{aligned} &\min\ f(x_k+\alpha d_k), \\ &\text{s.t.}\ \ 0 \le \alpha \le \bar{\alpha}, \end{aligned} \tag{10.22}$$

其中

$$\bar{\alpha} = \sup\{\alpha \,|\, g_i(x_k+\alpha d_k) \ge 0,\ i=1,\cdots,m\}. \tag{10.23}$$

现在, 我们写出求解问题 (10.17) 的可行方向法的详细计算步骤:

算法 26 (非线性约束的可行方向法)

步 1　确定 x_k 处的有效约束指标集 $I(x_k)$:

$$I(x_k) = \{i \,|\, g_i(x_k)=0,\ 1 \le i \le m\}.$$

若 $I(x_k)=\emptyset$ 且 $\|\nabla f(x_k)\| < \varepsilon_1$, 停算, 得到近似极小点 x_k; 否则若 $I(x_k)=\emptyset$ 但 $\|\nabla f(x_k)\| \ge \varepsilon_1$, 则取搜索方向 $d_k=-\nabla f(x_k)$, 转步 4. 反之, 若 $I(x_k)\neq\emptyset$, 转步 2.

步 2　求解线性规划问题 (10.20), 得最优解 d_k, 最优值 z_k.

步 3　若 $|z_k| < \varepsilon_2$, 停算, 输出 x_k 作为近似极小点. 否则, 以 d_k 作为搜索方向, 转步 4.

步 4　首先由 (10.23) 计算 $\bar{\alpha}$, 然后作一维线搜索:

$$\begin{cases} \min & f(x_k+\alpha d_k), \\ \text{s.t.} & 0 \le \alpha \le \bar{\alpha}, \end{cases}$$

求得最优解 α_k.

步 5　置 $x_{k+1} := x_k + \alpha_k d_k$, $k := k+1$, 转步 1.

注 (1) 步 1 中的 $I(x_k)=\emptyset$, 表明 x_k 是可行域 $\mathcal{F}$ 的内点, 因此任意方向都是可行方向. 此时, 若不满足终止条件, 类似于无约束优化问题, 可用最速下降法寻求下一个迭代点. 但毕竟不是真正的无约束问题, 步长要受到可行与便捷的限制.

(2) 步 3 中若 $z_k \approx 0$, 说明在 x_k 处找不到下降可行方向, 可以认为 x_k 是原问题的一个 Fritz-John 点.

(3) 算法 26 若推广到包含非线性等式约束的优化问题, 迭代过程会出现一些困难. 因为对于等式约束和当前可行迭代点 x_k, 一般难于找到一个可行方向. 这与罚函数类算法刚好相反, 罚函数类算法容易处理等式约束.

例 29 用 Zoutendijk 可行方向法求解下列问题:

$$\begin{aligned} \min\ & x_1^2 + x_2^2 - 2x_1 - 4x_2 + 3, \\ \text{s.t.}\ & -2x_1 + x_2 \geq -1, \\ & -x_1 - x_2 \geq -2, \\ & x_1 \geq 0,\ x_2 \geq 0. \end{aligned}$$

解 取初始可行点 $x_0 = (0, 0)^T$.

第 1 次迭代. $\nabla f(x_0) = (-2, -4)^T$. 有效约束和非有效约束的系数矩阵和右端向量分别为:

$$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -2 & 1 \\ -1 & -1 \end{bmatrix}, \quad b_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad b_2 = \begin{bmatrix} -1 \\ -2 \end{bmatrix}.$$

先求在 x_0 处的下降可行方向 d_0. 解线性规划问题

$$\begin{aligned} \min\ & \nabla f(x_0)^T d, \\ \text{s.t.}\ & A_1 d \geq 0, \\ & |d_i| < 1,\ i = 1, 2, \end{aligned}$$

即

$$\begin{aligned} \min\ & -2d_1 - 4d_2, \\ \text{s.t.}\ & d_1 \geq 0,\ d_2 \geq 0, \\ & -1 \leq d_1 \leq 1, \\ & -1 \leq d_2 \leq 1. \end{aligned}$$

由单纯形方法求得最优解为 $d_0 = (1, 1)^T$.

再求步长因子 α_0. 分别计算 $\bar{b}$, $\bar{d}$ 和 $\bar{\alpha}$:

$$\bar{d} = A_2 d_0 = \begin{bmatrix} -2 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \end{bmatrix},$$

$$\bar{b} = b_2 - A_2 x_0 = \begin{bmatrix} -1 \\ -2 \end{bmatrix} - \begin{bmatrix} -2 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \end{bmatrix},$$

$$\bar{\alpha} = \min\left\{\frac{-1}{-1}, \frac{-2}{-2}\right\} = 1.$$

于是解下面的一维搜索问题求 α_0:

$$\begin{aligned} &\min\ f(x_0 + \alpha d_0) = 2\alpha^2 - 6\alpha + 3, \\ &\text{s.t. } 0 \le \alpha \le 1. \end{aligned}$$

求得 $\alpha_0 = 1$. 令 $x_1 = x_0 + \alpha_0 d_0 = (1, 1)^T$. 至此第 1 次迭代完成.

第 2 次迭代. $\nabla f(x_1) = (0, -2)^T$. 有效约束和非有效约束的系数矩阵和右端向量分别为:

$$A_1 = \begin{bmatrix} -2 & 1 \\ -1 & -1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad b_1 = \begin{bmatrix} -1 \\ -2 \end{bmatrix}, \quad b_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

解线性规划问题

$$\begin{aligned} &\min\ -2d_2, \\ &\text{s.t. } 2d_1 + d_2 \ge 0, \\ &\qquad -d_1 - d_2 \ge 0, \\ &\qquad -1 \le d_1 \le 1, \\ &\qquad -1 \le d_2 \le 1. \end{aligned}$$

由单纯形方法求得最优解为 $d_1 = (-1, 1)^T$.

再求步长因子 α_1. 分别计算 $\bar{b}$, $\bar{d}$ 和 $\bar{\alpha}$:

$$\bar{d} = A_2 d_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix},$$

$$\bar{b} = b_2 - A_2 x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix},$$

得 $\bar{\alpha} = 1$. 于是解下面的一维搜索问题求 α_1:

$$\begin{aligned} &\min\ f(x_1 + \alpha d_1) = 2\alpha^2 - 2\alpha - 1, \\ &\text{s.t. } 0 \le \alpha \le 1. \end{aligned}$$

求得 $\alpha_1 = 0.5$. 令 $x_2 = x_1 + \alpha_1 d_1 = (1,1)^T + 0.5(-1,1)^T = (0.5, 1.5)^T$. 至此第 2 次迭代完成.

第 3 次迭代. $\nabla f(x_2) = (-1,-1)^T$. 有效约束和非有效约束的系数矩阵和右端向量分别为:

$$A_1 = \begin{bmatrix} -1 & -1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -2 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad b_1 = \begin{bmatrix} -2 \end{bmatrix}, \quad b_2 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}.$$

解线性规划问题

$$\begin{aligned} &\min\ -d_1 - d_2, \\ &\text{s.t. } -d_1 - d_2 \ge 0, \\ &\qquad -1 \le d_1 \le 1, \\ &\qquad -1 \le d_2 \le 1. \end{aligned}$$

由单纯性方法求得最优解为 $d_1 = (0,0)^T$. 由定理 49 知, $x_2 = (0.5, 1.5)^T$ 是 KT 点. 由于此例是凸规划, 因此, x_2 也是最优解, 相应地, 目标函数的最优值为 $f_{\min} = f(x_2) = 1.5$.

10.2　梯度投影法

梯度投影法是 Rosen 于 1961 年针对线性约束的优化问题首先提出来的一种优化算法, 次年 Rosen 又将他的这一算法推广到处理非线性约束的情形. 后来这

一方法又得到了进一步的发展, 成为求解非线性规划问题的一类重要的方法.

10.2.1　梯度投影法的理论基础

我们考虑线性约束的优化问题

$$\begin{cases} \min f(x), \\ \text{s.t.}\ \ Ax \geq b, \\ \qquad Ex = e, \end{cases} \tag{10.24}$$

其中 f 是连续可微的 n 元实函数, $A \in \mathbb{R}^{m\times n}$, $E \in \mathbb{R}^{l\times n}$, $b \in \mathbb{R}^m$, $e \in \mathbb{R}^l$, $x \in \mathbb{R}^n$. 其可行域为 $\mathcal{D} = \{x \in \mathbb{R}^n \mid Ax \geq b,\ Ex = e\}$.

梯度投影法的基本思想是: 当迭代点 x_k 是可行域 $\mathcal{D}$ 的内点时, 取 $d = -\nabla f(x_k)$ 作为搜索方向; 否则, 当 x_k 是可行域 $\mathcal{D}$ 的边界点时, 取 $-\nabla f(x_k)$ 在这些边界面交集上的投影作为搜索方向. 这也是“梯度投影法”名称的由来.

在具体介绍梯度投影法之前, 我们先引入投影矩阵的概念及其有关性质.

定义 20　称矩阵 $P \in \mathbb{R}^{n\times n}$ 为投影矩阵, 是指 P 满足

$$P = P^T, \quad P^2 = P.$$

由上述定义可知, 一个对称幂等矩阵就是投影矩阵. 投影矩阵具有如下一些基本性质, 其证明可参看文献 [11].

引理 20　设矩阵 $P \in \mathbb{R}^{n\times n}$.

(1) 若 P 为投影矩阵, 则 P 是半正定的. (2) P 是投影矩阵当且仅当 $I-P$ 也是投影矩阵, 其中 I 是 n 阶单位阵. (3) 设 P 是投影矩阵, $Q = I - P$, 则

$$L = \{y = Px | x \in \mathbb{R}^n\}, \quad L^{\perp} = \{z = Qx | x \in \mathbb{R}^n\}$$

是互相正交的线性子空间, 并且对于任意的 $x \in \mathbb{R}^n$ 可唯一地表示为

$$x = y + z,\ y \in L,\ z \in L^{\perp}.$$

定理 51　设 $\bar{x}$ 是问题 (10.24) 的一个可行点, 且满足 $A_1\bar{x} = b_1$, $A_2x > b_2$, 其中

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

又设

$$M=\begin{bmatrix}A_1\\E\end{bmatrix}$$

是满秩矩阵, $P=I-M^T(MM^T)^{-1}M$, $P\nabla f(\bar{x})\neq 0$. 若取 $d=-P\nabla f(\bar{x})$, 则 d 是问题 (10.24) 的一个下降可行方向.

证 不难验证, $P=I-M^T(MM^T)^{-1}M$ 是投影矩阵, 故由 $P\nabla f(\bar{x})\neq 0$ 可得

$$\nabla f(\bar{x})^Td=-\nabla f(\bar{x})^TP\nabla f(\bar{x})^T=-\|P\nabla f(\bar{x})^T\|^2<0,$$

即 d 为下降方向. 又因

$$\begin{aligned}Md&=-MP\nabla f(\bar{x})^T\\&=-M(I-M^T(MM^T)^{-1}M)\nabla f(\bar{x})^T\\&=(-M+M)\nabla f(\bar{x})^T=0,\end{aligned}$$

即

$$Md=\begin{bmatrix}A_1\\E\end{bmatrix}d=\begin{bmatrix}A_1d\\Ed\end{bmatrix}=0,$$

从而 $A_1d=0$, $Ed=0$. 根据引理 17, d 是在 $\bar{x}$ 的可行方向. 证毕. □

上述定理在 $P\nabla f(\bar{x})\neq 0$ 的假设下, 给出了用投影来求下降可行方向的一种方法. 但是, 当 $P\nabla f(\bar{x})=0$ 时, 情况又该如何呢? 下面的定理指出, 此时有两种可能: 要么 $\bar{x}$ 是已是 KT 点, 要么构造新的投影矩阵, 以便求得下降可行方向.

定理 52 设 $\bar{x}$ 是问题 (10.24) 的一个可行点, 且满足 $A_1\bar{x}=b_1$, $A_2x>b_2$, 其中

$$A=\begin{bmatrix}A_1\\A_2\end{bmatrix},\quad b=\begin{bmatrix}b_1\\b_2\end{bmatrix}.$$

又设

$$M=\begin{bmatrix}A_1\\E\end{bmatrix}$$

是行满秩矩阵, 令

$$P = I - M^T(MM^T)^{-1}M,$$

$$\omega = (MM^T)^{-1}M\nabla f(\bar{x}) = \begin{bmatrix} \lambda \\ \mu \end{bmatrix},$$

其中 λ 和 μ 分别对应于 A_1 和 E. 若 $P\nabla f(\bar{x}) = 0$, 则

(1) 如果 $\lambda \geq 0$, 那么 $\bar{x}$ 是 KT 点.

(2) 如果 $\lambda \ngeq 0$, 不妨设 $\lambda_j < 0$, 那么先从 A_1 中去掉 λ_j 所对应的行, 得到新矩阵 $\tilde{A}_1$, 然后令

$$\tilde{M} = \begin{bmatrix} \tilde{A}_1 \\ E \end{bmatrix}, \quad \tilde{P} = I - \tilde{M}^T(\tilde{M}\tilde{M}^T)^{-1}\tilde{M}, \quad d = -\tilde{P}\nabla f(\bar{x}),$$

那么, d 是下降可行方向.

证 (1) 设 $\lambda \geq 0$. 注意到 $P\nabla f(\bar{x}) = 0$, 我们有

$$\begin{aligned} 0 &= P\nabla f(\bar{x}) = [I - M^T(MM^T)^{-1}M]\nabla f(\bar{x}) \\ &= \nabla f(\bar{x}) - M^T(MM^T)^{-1}M\nabla f(\bar{x}) \\ &= \nabla f(\bar{x}) - M^T\omega = \nabla f(\bar{x}) - [A_1^T, E^T]\begin{bmatrix} \lambda \\ \mu \end{bmatrix} \\ &= \nabla f(\bar{x}) - A_1^T\lambda - E^T\mu. \end{aligned} \tag{10.25}$$

(10.25) 恰为 KT 条件, 因此 $\bar{x}$ 是 KT 点.

(2) 设 $\lambda_j < 0$. 先证明 $\tilde{P}\nabla f(\bar{x}) \neq 0$. 用反证法. 若 $\tilde{P}\nabla f(\bar{x}) = 0$, 则由 $\tilde{P}$ 定义可得

$$0 = \tilde{P}\nabla f(\bar{x}) = [I - \tilde{M}^T(\tilde{M}\tilde{M}^T)^{-1}\tilde{M}]\nabla f(\bar{x}) = \nabla f(\bar{x}) - \tilde{M}^T\tilde{\omega}, \tag{10.26}$$

其中 $\tilde{\omega} = (\tilde{M}\tilde{M}^T)^{-1}\tilde{M}\nabla f(\bar{x})$. 设 A_1 中对应于 λ_j 的行为 r_j (第 j 行). 由于

$$A_1^T\lambda + E^T\mu = \tilde{A}_1^T\tilde{\lambda} + \lambda_j r_j^T + E^T\mu = \tilde{M}^T\bar{\omega} + \lambda_j r_j^T. \tag{10.27}$$

将 (10.27) 代入 (10.25) 得

$$0 = \nabla f(\bar{x}) - \tilde{M}^T\bar{\omega} - \lambda_j r_j^T. \tag{10.28}$$

(10.26) 减去 (10.28) 得

$$\tilde{M}^T(\bar{\omega}-\tilde{\omega})+\lambda_j r_j=0. \tag{10.29}$$

上式左端是矩阵 M 的行向量的一个线性组合, 且至少有一个系数 $\lambda_j\neq 0$. 由此可得 M 的行向量线性相关, 这与 M 是行满秩矛盾. 因此必有 $\tilde{P}\nabla f(\bar{x})\neq 0$.

由于 $\tilde{P}$ 亦为投影矩阵, 且 $\tilde{P}\nabla f(\bar{x})\neq 0$, 故

$$\nabla f(\bar{x})^T d=-\nabla f(\bar{x})^T\tilde{P}\nabla f(\bar{x})^T=-\|\tilde{P}\nabla f(\bar{x})\|^2<0,$$

即 d 是下降方向. 以下只需证明 d 是可行方向即可. 事实上, 因

$$\begin{aligned}\tilde{M}d &= -\tilde{M}\tilde{P}\nabla f(\bar{x})\\ &= -\tilde{M}[I-\tilde{M}^T(\tilde{M}\tilde{M}^T)^{-1}\tilde{M}]\nabla f(\bar{x})\\ &= -(\tilde{M}-\tilde{M})\nabla f(\bar{x})=0,\end{aligned}$$

即

$$\tilde{A}_1 d=0,\quad Ed=0. \tag{10.30}$$

将 (10.28) 两边左乘 $r_j\tilde{P}$ 得

$$r_j\tilde{P}\nabla f(\bar{x})-r_j\tilde{P}\tilde{M}^T\bar{\omega}-\lambda_j r_j\tilde{P}r_j^T=0.$$

注意到 $\tilde{P}\tilde{M}^T=0$ 及 $d=-\tilde{P}\nabla f(\bar{x})$, 上式即

$$r_j d+\lambda_j r_j\tilde{P}r_j^T=0. \tag{10.31}$$

因 $\tilde{P}$ 半正定 $(r_j\tilde{P}r_j^T\geq 0)$ 及 $\lambda_j<0$, 故有

$$r_j d=-\lambda_j r_j\tilde{P}r_j^T\geq 0. \tag{10.32}$$

由 (10.30) 和 (10.32) 即得

$$A_1 d\geq 0,\quad Ed=0.$$

最后, 根据引理 17, d 是在 $\bar{x}$ 的可行方向. 证毕. □

10.2.2　梯度投影法的计算步骤

基于上述分析与讨论，我们给出 Rosen 梯度投影法的详细计算步骤如下:

算法 27 (Rosen **梯度投影法**)

步 0　给定初始可行点 x_0, 令 $k := 0$.

步 1　在 x_k 处确定有效约束 $A_1x_k = b_1$ 和非有效约束 $A_2x_k > b_2$, 其中

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

步 2　令

$$M = \begin{bmatrix} A_1 \\ E \end{bmatrix}.$$

若 M 是空的, 则令 $P = I$ (单位矩阵). 否则, 令 $P = I - M^T(MM^T)^{-1}M$.

步 3　计算 $d_k = -P\nabla f(x_k)$. 若 $\|d_k\| \neq 0$, 转步 5; 否则, 转步 4.

步 4　计算

$$\omega = (MM^T)^{-1}M\nabla f(x_k) = \begin{bmatrix} \lambda \\ \mu \end{bmatrix}.$$

若 $\lambda \geq 0$, 停算, 输出 x_k 为 KT 点. 否则, 选取 λ 的某个负分量, 比如 $\lambda_j < 0$, 修正矩阵 A_1, 即去掉 A_1 中对应于 λ_j 的行, 转步 2.

步 5　求解一维搜索问题, 确定步长 α_k :

$$\begin{aligned} &\min\ f(x_k + \alpha d_k), \\ &\text{s.t.}\ \ 0 \leq \alpha \leq \bar{\alpha}, \end{aligned}$$

其中 $\bar{\alpha}$ 由下式确定

$$\bar{\alpha} = \begin{cases} \min\left\{ \dfrac{(b_2 - A_2x_k)_i}{(A_2d_k)_i} \middle| (A_2d_k)_i < 0 \right\}, & (A_2d_k) \not\geq 0, \\ +\infty, & (A_2d_k) \geq 0. \end{cases}$$

步 6　令 $x_{k+1} := x_k + \alpha_k d_k$, $k := k + 1$, 转步 1.

例 30　用 Rosen 梯度投影法求解下面的优化问题

$$\begin{cases} \min f(x) = x_1^2 + x_2^2 + 6x_1 + 9, \\ \text{s.t. } 2x_1 + x_2 \geq 4, \\ \qquad x_1, x_2 \geq 0. \end{cases}$$

取初始可行点为 $x_0 = (2, 0)^T$.

解　首先目标函数的梯度

$$\nabla f(x) = \begin{bmatrix} 2x_1 + 6 \\ 2x_2 \end{bmatrix}.$$

第 1 次迭代. 在 x_0 处的梯度为

$$\nabla f(x_0) = \begin{bmatrix} 10 \\ 0 \end{bmatrix}.$$

在 x_0 处有效约束的指标集为 $I = \{1, 3\}$, 即 $2x_1 + x_2 \geq 4$ 和 $x_2 \geq 0$ 是在 $x_0 = (2, 0)^T$ 处的有效约束. 将约束系数矩阵 A 和右端向量 b 分解为

$$A_1 = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}, \quad b_1 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad b_2 = \begin{bmatrix} 0 \end{bmatrix}.$$

因 $E = \emptyset$, 故 $M = A_1$. 计算投影矩阵

$$\begin{aligned} P &= I - A_1^T (A_1 A_1^T)^{-1} A_1 \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} \left(\begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} \left(\frac{1}{4} \begin{bmatrix} 1 & -1 \\ -1 & 5 \end{bmatrix} \right) \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

令

$$d_0 = -P \nabla f(x_0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

计算

$$\begin{aligned}\lambda &= (A_1A_1^T)^{-1}A_1\nabla f(x_0)\\ &= \left(\frac{1}{4}\begin{bmatrix}1 & -1\\ -1 & 5\end{bmatrix}\right)\begin{bmatrix}2 & 1\\ 0 & 1\end{bmatrix}\begin{bmatrix}10\\ 0\end{bmatrix} = \begin{bmatrix}5\\ -5\end{bmatrix} = \begin{bmatrix}\lambda_1\\ \lambda_2\end{bmatrix}.\end{aligned}$$

修正 A_1. 去掉 A_1 中对应 $\lambda_2 = -5$ 的行, 得到

$$\tilde{A}_1 = [2,\ 1].$$

再求投影矩阵

$$\begin{aligned}\tilde{P} &= I - \tilde{A}_1^T(\tilde{A}_1\tilde{A}_1^T)^{-1}\tilde{A}_1\\ &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - \begin{bmatrix}2\\ 1\end{bmatrix}\left(\begin{bmatrix}2 & 1\end{bmatrix}\begin{bmatrix}2\\ 1\end{bmatrix}\right)^{-1}\begin{bmatrix}2 & 1\end{bmatrix}\\ &= \frac{1}{5}\begin{bmatrix}1 & -2\\ -2 & 4\end{bmatrix}.\end{aligned}$$

令

$$\tilde{d}_0 = -\tilde{P}\nabla f(x_0) = -\frac{1}{5}\begin{bmatrix}1 & -2\\ -2 & 4\end{bmatrix}\begin{bmatrix}10\\ 0\end{bmatrix} = \begin{bmatrix}-2\\ 4\end{bmatrix}.$$

求步长 α_0:

$$\begin{aligned}&\min\ f(x_0 + \alpha\tilde{d}_0),\\ &\text{s.t.}\ \ 0 \le \alpha \le \bar{\alpha}.\end{aligned} \tag{10.33}$$

由于

$$\begin{aligned}\hat{b} &= b_2 - A_2x_0 = 0 - \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}2\\ 0\end{bmatrix} = -2.\\ \hat{d} &= A_2\tilde{d}_0 = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}-2\\ 4\end{bmatrix} = -2.\end{aligned}$$

故

$$\bar{\alpha} = \frac{-2}{-2} = 1.$$

这样, 问题 (10.33) 即为

$$\begin{aligned}&\min\ 20\alpha^2-20\alpha+25,\\ &\text{s.t.}\ \ 0\le\alpha\le 1.\end{aligned}$$

解之得 $\alpha_0=\dfrac{1}{2}=0.5$. 从而

$$x_1=x_0+\alpha_0 d_0=\begin{bmatrix}2\\0\end{bmatrix}+\frac{1}{2}\begin{bmatrix}-2\\4\end{bmatrix}=\begin{bmatrix}1\\2\end{bmatrix}.$$

第 2 次迭代.

$$\nabla f(x_1)=\begin{bmatrix}2\times 1+6\\ 2\times 2\end{bmatrix}=\begin{bmatrix}8\\4\end{bmatrix}.$$

$$A_1=\begin{bmatrix}2 & 1\end{bmatrix},\quad b_1=\begin{bmatrix}4\end{bmatrix},\quad A_2=\begin{bmatrix}1&0\\0&1\end{bmatrix},\quad b_2=\begin{bmatrix}0\\0\end{bmatrix}.$$

因 $E=\emptyset$, 故 $M=A_1$. 计算投影矩阵

$$P=I-A_1^T(A_1A_1^T)^{-1}A_1=\frac{1}{5}\begin{bmatrix}1&-2\\-2&4\end{bmatrix}.$$

令

$$d_2=-P\nabla f(x_1)=-\frac{1}{5}\begin{bmatrix}1&-2\\-2&4\end{bmatrix}\begin{bmatrix}8\\4\end{bmatrix}=\begin{bmatrix}0\\0\end{bmatrix}.$$

计算

$$\lambda=(A_1A_1^T)^{-1}A_1\nabla f(x_1)=\frac{1}{5}\begin{bmatrix}2&1\end{bmatrix}\begin{bmatrix}8\\4\end{bmatrix}=4>0,$$

故 $x_1=(1,2)^T$ 是 KT 点. □

10.3 简约梯度法

10.3.1 Wolfe 简约梯度法

Wolfe 于 1963 年针对线性等式约束的非线性优化问题提出一种新的可行方向法, 称之为简约梯度法. 我们来介绍这种方法.

考虑具有约性约束的非线性优化问题

$$\begin{cases} \min\ f(x), \\ \text{s.t. } Ax = b, \\ \qquad x \geq 0, \end{cases} \tag{10.34}$$

其中 $A \in \mathbb{R}^{m\times n}$, 秩为 m, $b \in \mathbb{R}^m$, $f : \mathbb{R}^n \to \mathbb{R}$ 连续可微. 设矩阵 A 的任意 m 个列都线性无关，并且约束条件的每个基本可行点都有 m 个正分量. 那么在此假设下, 每个可行解至少有 m 个正分量, 至多有 $n-m$ 个零分量. 简约梯度法的基本思想把求解线性规划的单纯形法推广到解线性约束的非线性优化问题 (10.34). 先利用等是约束条件消去一些变量，然后利用降维所形成的简约梯度来构造下降方法, 接着作线性搜索求步长, 重复此过程逐步逼近极小点. 下面依次介绍如何确立简约梯度、如何构造下降方向和计算线搜索的步长上界等.

先介绍简约梯度的确立. 将 A 和 x 进行分解. 不失一般性, 可令

$$A = [B,\ N], \quad x = \begin{bmatrix} x_B \\ x_N \end{bmatrix},$$

其中 B 是 $m\times m$ 可逆矩阵, x_B, x_N 分别是由基变量和非基变量构成的向量. 那么线性约束 $Ax = b$ 就可以表示为

$$Bx_B + Nx_N = b,$$

而 $x \geq 0$ 则变成

$$x_B = B^{-1}b - B^{-1}Nx_N \geq 0, \quad x_N \geq 0.$$

现假设 x 是非退化的可行解, 即 $x_B > 0$. 由于 x_B 可以用 x_N 来表示, 因此 $f(x)$ 可以化成关于 x_N 的函数, 即

$$f(x) = f(x_B, x_N) = f(B^{-1}b - B^{-1}Nx_N, x_N) := F(x_N).$$

称 $n-m$ 维向量 x_N 的函数 $F(x_N)$ 的梯度为 $f(x)$ 的简约梯度, 记为 $r(x_N)$, 即

$$\begin{aligned} r(x_N) &= \nabla_{x_N}F(x_N) = \nabla_{x_N}f(B^{-1}b - B^{-1}Nx_N,\ x_N) \\ &= \nabla_N f(x_B, x_N) - (B^{-1}N)^T\nabla_B f(x_B, x_N), \end{aligned} \tag{10.35}$$

上式中 $\nabla_N = \nabla_{x_N}$, $\nabla_B = \nabla_{x_B}$.

再确定搜索方向, 即怎样确定在点 x_k 处的下降可行方向 d_k, 使得后继点 $x_{k+1} + \alpha_k d_k$ 是可行点且目标函数值下降. 令

$$d_k = \begin{bmatrix} d_k^B \\ d_k^N \end{bmatrix}.$$

欲使 d_k 为下降可行方向, 需其满足

$$\begin{cases} \nabla f(x_k)^T d_k < 0, \\ Ad_k = 0, \\ (d_k)_j \geq 0, \quad 若\ (x_k)_j = 0. \end{cases}$$

由等式 $Ad_k = 0$ 得

$$Bd_k^B + Nd_k^N = 0,$$

即

$$d_k^B = -B^{-1}Nd_k^N, \tag{10.36}$$

这表明 d_k^B 是由 d_k^N 确定的. 再由下降性条件 $\nabla f(x_k)^T d_k < 0$ 可得

$$\begin{aligned} \nabla f(x_k)^T d_k &= \nabla_B f(x_k)^T d_k^B + \nabla_N f(x_k)^T d_k^N \\ &= -\nabla_B f(x_k)^T B^{-1}Nd_k^N + \nabla_N f(x_k)^T d_k^N \\ &= r(x_k^N)^T d_k^N < 0. \end{aligned} \tag{10.37}$$

由非负性条件可知

$$当\ (x_k^N)_j = 0\ 时,\ \ (d_k^N)_j \geq 0. \tag{10.38}$$

不难发现, 满足 (10.36)~(10.38) 的 d_k 有许多种选取方法, 其中一种简单的取法为

$$(d_k^N)_j = \begin{cases} -(x_k^N)_j\, r_j(x_k^N), & 如果\ r_j(x_k^N) \geq 0 \\ -r_j(x_k^N), & 否则. \end{cases} \tag{10.39}$$

$$d_k = \begin{bmatrix} -B^{-1}Nd_k^N \\ d_k^N \end{bmatrix} = \begin{bmatrix} -B^{-1}N \\ I_{n-m} \end{bmatrix} d_k^N \tag{10.40}$$

.

余下的问题就是确定步长 α_k. 为保持

$$x_{k+1} \geq 0,$$

即

$$(x_{k+1})_j = (x_k)_j + \alpha(d_k)_j \geq 0, \quad j = 1\cdots,n, \tag{10.41}$$

需确定 α 的取值范围.

注意到, 当 $(d_k)_j \geq 0$ 时, 对任意的 $\alpha \geq 0$, (10.41) 恒成立. 而当 $(d_k)_j < 0$ 时, 应取

$$\alpha \leq \frac{(x_k)_j}{-(d_k)_j}.$$

因此, 令

$$\bar{\alpha} = \begin{cases} +\infty, & d_k \geq 0 \\ \min\left\{ -\dfrac{(x_k)_j}{(d_k)_j} \middle| (d_k)_j < 0 \right\}, & \text{否则}. \end{cases} \tag{10.42}$$

可以证明, 按照上述方式构造的方向 d_k, 若 $d_k \neq 0$, 则它必为下降可行方向, 否则, 相应的 x_k 必为 KT 点.

定理 53 设 $A = (B,N)$ 是 $m \times n$ 矩阵, B 是 m 阶非奇异矩阵, $x_k = \begin{bmatrix} x_k^B \\ x_k^N \end{bmatrix}$ 是问题 (10.34) 的可行点, 其中 $x_k^B > 0$ 是相应于 B 的 m 维向量. 又假定函数 f 在 x_k 处连续可微, d_k 是由 (10.39) 和 (10.40) 定义的方向向量. 那么, 若 $d_k \neq 0$, 则 d_k 是下降可行方向, 且 $d_k = 0$ 的充要条件是 x_k 是 KT 点.

证 由 d_k 的定义, 有

$$Ad_k = Bd_k^B + Nd_k^N = B(-B^{-1}Nd_k^N) + Nd_k^N = 0.$$

又由 (10.38), 当 $(x_N)_j = 0$ 时, $(d_k)_j \geq 0$. 注意到 $x_k^B > 0$, 因此根据引理 17, d_k 是可行方向. 此外, 我们有

$$\begin{aligned} \nabla f(x_k)^T d_k &= \nabla_B f(x_k)^T d_k^B + \nabla_N f(x_k)^T d_k^N \\ &= \nabla_B f(x_k)^T (-B^{-1}Nd_k^N) + \nabla_N f(x_k)^T d_k^N \\ &= r(x_k^N)^T d_k. \end{aligned}$$

注意到, 当 $d_k^N \neq 0$ 时, 根据 (10.39) 知 $r(x_k^N)^T d_k < 0$, 因此 d_k 是下降可行方向. 现在证明 $d_k = 0$ 当且仅当 x_k 是 KT 点. 事实上, 我们知道, x_k 是问题 (10.34) 的 KT 点的充要条件是, 存在乘子 $\lambda = (\lambda_B^T, \lambda_N^T)^T \geq 0$ 及 μ, 使得

$$\nabla f(x_k) - A^T\mu - \lambda = 0,$$
$$\lambda \geq 0,\ \ x_k \geq 0,\ \ \lambda^T x = 0,$$

即

$$\begin{bmatrix} \nabla_B f(x_k) \\ \nabla_N f(x_k) \end{bmatrix} - \begin{bmatrix} B^T \\ N^T \end{bmatrix} \mu - \begin{bmatrix} \lambda_B \\ \lambda_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{10.43}$$

$$\lambda_B^T x_k^B = 0, \quad \lambda_N^T x_k^N = 0. \tag{10.44}$$

我们先设 x_k 是 KT 点, 则上述条件成立. 由于 $x_k^B > 0$ 且 $\lambda_B \geq 0$, 则由 (10.44) 的第一式推出 $\lambda_B = 0$. 从而由 (10.43) 的第一个方程可得

$$\mu = (B^T)^{-1}\nabla_B f(x_k). \tag{10.45}$$

将上式代入 (10.43) 的第二个方程可求得

$$\lambda_N = \nabla_N f(x_k) - (B^{-1}N)^T \nabla_B f(x_k) = r(x_k^N) \geq 0. \tag{10.46}$$

由 (10.46) 及 (10.44) 的第二式可得

$$r(x_k^N)^T x_k^N = 0. \tag{10.47}$$

注意到 $x_k^N \geq 0$, 故由上式可推出

$$r_j(x_k^N)(x_k^N)_j = 0,\ \ j = 1, \cdots, n. \tag{10.48}$$

因此, 根据 (10.46), (10.48) 和 (10.36), (10.39) 可推得 $d_k = 0$.

反之, 设 $d_k = 0$, 则 $r(x_k^N)$ 均非负. 令

$$\lambda_N = r(x_k^N) = \nabla_N f(x_k) - (B^{-1}N)^T \nabla_B f(x_k) \geq 0.$$

故由 (10.39) 可知必有 $\lambda_N^T x_k^N = 0$ 成立. 再令

$$\lambda_B = 0,\ \ \mu = (B^T)^{-1}\nabla_B f(x_k),$$

则有 $\lambda_B^T x_k^B = 0$ 及 (10.43) 式成立. 故 x_k 是 KT 点. 证毕. □

下面, 我们写出简约梯度法的详细计算步骤如下:

算法 28 (Wolfe 简约梯度法)

步 0 初始化. 选取初始可行点 $x_0 \in \mathbb{R}^n$. 令 $k := 0$.

步 1 计算搜索方向. 将 x_k 分解成

$$x_k = \begin{bmatrix} x_k^B \\ x_k^N \end{bmatrix},$$

其中, x_k^B 为基变量, 由 x_k 的 m 个最大分量组成, 这些分量的下标集记作 J_k. 相应地, 将 A 分解成 $A = (B, N)$. 按下式计算 d_k :

$$\begin{aligned} r(x_k^N) &= \nabla_N f(x_k^B, x_k^N) - (B^{-1}N)^T \nabla_B f(x_k^B, x_k^N), \\ (d_k^N)_j &= \begin{cases} -(x_k^N)_j \, r_j(x_k^N), & \text{如果 } r_j(x_k^N) \geq 0, \\ -r_j(x_k^N), & \text{否则}, \end{cases} \\ d_k &= \begin{bmatrix} d_k^B \\ d_k^N \end{bmatrix} = \begin{bmatrix} -B^{-1}N \\ I_{n-m} \end{bmatrix} d_k^N. \end{aligned}$$

步 2 检验终止准则. 若 $d_k = 0$, 则 x_k 为 KT 点, 停算. 否则, 转步 3.

步 3 计算步长上界 $\bar{\alpha}$:

$$\bar{\alpha} = \begin{cases} +\infty, & d_k \geq 0 \\ \min\left\{ -\dfrac{(x_k)_j}{(d_k)_j} \middle| (d_k)_j < 0 \right\}, & \text{否则}. \end{cases}$$

步 4 进行一维搜索. 求解下面的一维极小化问题得步长 α_k :

$$\begin{cases} \min f(x_k + \alpha d_k), \\ \text{s.t. } 0 \leq \alpha \leq \bar{\alpha}. \end{cases}$$

令 $x_{k+1} := x_k + \alpha_k d_k$.

步 5 修正基变量. 若 $x_{k+1}^B > 0$, 则基变量不变. 否则, 若有 j 使得 $(x_{k+1})_j = 0$, 则将 $(x_{k+1})_j$ 换出基, 而以 $(x_{k+1}^N)_j$ 中最大分量换入基, 构成新的基向量 x_{k+1}^B 和 x_{k+1}^N.

步 6 令 $k := k + 1$, 转步 1.

例 31　用 Wolfe 简约梯度法重新求解例 29, 即

$$\begin{aligned}
&\min\ x_1^2+x_2^2-2x_1-4x_2+3,\\
&\text{s.t. } -2x_1+x_2\geq -1,\\
&\qquad -x_1-x_2\geq -2,\\
&\qquad x_1\geq 0,\ x_2\geq 0.
\end{aligned}$$

解　首先, 引入松弛变量 $x_3,x_4\geq 0$, 将原问题转化为等价的“标准形式”:

$$\begin{aligned}
&\min\ x_1^2+x_2^2-2x_1-4x_2+3,\\
&\text{s.t.}\begin{cases}
2x_1-x_2+x_3=1,\\
x_1+x_2+x_4=2,\\
x_1\geq 0,\ x_2\geq 0,\\
x_3\geq 0,\ x_4\geq 0.
\end{cases}
\end{aligned}$$

首先写出约束矩阵和梯度

$$A=\begin{bmatrix}2&-1&1&0\\1&1&0&1\end{bmatrix},\quad \nabla f(x)=\begin{bmatrix}2x_1-2\\2x_2-4\\0\\0\end{bmatrix}.$$

取初始可行点 $x_0=(0,0,1,2)^T$.

第 1 次迭代. $k=0$. $J_0=\{3,4\}$. $\nabla f(x_0)=(-2,-4,0,0)^T$. 确立 B,N 等得

$$x_0^B=\begin{bmatrix}1\\2\end{bmatrix},\quad x_0^N=\begin{bmatrix}0\\0\end{bmatrix},\quad B=\begin{bmatrix}1&0\\0&1\end{bmatrix},\quad N=\begin{bmatrix}2&-1\\1&1\end{bmatrix}.$$

计算

$$B^{-1}N=\begin{bmatrix}2&-1\\1&1\end{bmatrix},\quad \nabla_N f(x_0)=\begin{bmatrix}-2\\-4\end{bmatrix},\quad \nabla_B f(x_0)=\begin{bmatrix}0\\0\end{bmatrix}.$$

计算简约梯度 $r(x_0^N)$ 得

$$\begin{aligned} r(x_0^N) &= \nabla_N f(x_0) - (B^{-1}N)^T \nabla_B f(x_0) \\ &= \begin{bmatrix} -2 \\ -4 \end{bmatrix} - \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -2 \\ -4 \end{bmatrix}. \end{aligned}$$

当 $r_j(x_0^N) \le 0$ 时, 取 $(d_0^N)_j = -r_j(x_0^N)$, 故

$$d_0^N = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, \quad d_0^B = -B^{-1}Nd_0^N = -\begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ -6 \end{bmatrix}.$$

从而 $d_0 = (2, 4, 0, -6)^T$.

求步长上界

$$\bar{\alpha} = -\frac{2}{-6} = \frac{1}{3}.$$

从 x_0 出发, 沿 d_0 搜索:

$$x_0 + \alpha d_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 2 \end{bmatrix} + \alpha \begin{bmatrix} 2 \\ 4 \\ 0 \\ -6 \end{bmatrix} = \begin{bmatrix} 2\alpha \\ 4\alpha \\ 1 \\ 2-6\alpha \end{bmatrix},$$

$$f(x_0 + \alpha d_0) = 20\alpha^2 - 20\alpha + 3.$$

求解一维极小问题

$$\begin{aligned} &\min 20\alpha^2 - 20\alpha + 3, \\ &\text{s.t. } 0 \le \alpha \le \frac{1}{3}, \end{aligned}$$

得 $\alpha_0 = \frac{1}{3}$, 从而 $x_1 = x_0 + \alpha_0 d_0 = \left(\frac{2}{3}, \frac{4}{3}, 1, 0\right)^T$.

第 2 次迭代. $J_1 = \{2, 3\}$. $\nabla f(x_1) = \left(-\frac{2}{3}, -\frac{4}{3}, 0, 0\right)^T$.

$$x_1^B = \begin{bmatrix} \frac{4}{3} \\ 1 \end{bmatrix}, \quad x_1^N = \begin{bmatrix} \frac{2}{3} \\ 0 \end{bmatrix}, \quad B = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}, \quad N = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}.$$

计算

$$B^{-1}N=\begin{bmatrix}1&1\\3&1\end{bmatrix},\quad \nabla_N f(x_1)=\begin{bmatrix}-\frac{2}{3}\\0\end{bmatrix},\quad \nabla_B f(x_1)=\begin{bmatrix}-\frac{4}{3}\\0\end{bmatrix}.$$

计算简约梯度 $r(x_1^N)$ 得

$$\begin{aligned}r(x_1^N)&=\nabla_N f(x_1)-(B^{-1}N)^T\nabla_B f(x_1)\\&=\begin{bmatrix}-\frac{2}{3}\\0\end{bmatrix}-\begin{bmatrix}1&1\\3&1\end{bmatrix}^T\begin{bmatrix}-\frac{4}{3}\\0\end{bmatrix}=\begin{bmatrix}\frac{2}{3}\\\frac{4}{3}\end{bmatrix}.\end{aligned}$$

当 $r_j(x_1^N)>0$ 时, 取 $(d_1^N)_j=-(x_1^N)_j r_j(x_1^N)$, 故

$$d_1^N=\begin{bmatrix}-\frac{4}{9}\\0\end{bmatrix},\quad d_1^B=-(B^{-1}N)d_1^N=-\begin{bmatrix}1&1\\3&1\end{bmatrix}\begin{bmatrix}-\frac{4}{9}\\0\end{bmatrix}=\begin{bmatrix}\frac{4}{9}\\\frac{4}{3}\end{bmatrix}.$$

从而

$$d_1=\Big(-\frac{4}{9},\frac{4}{9},\frac{4}{3},0\Big)^T.$$

求步长上界

$$\bar{\alpha}=-\frac{2/3}{-4/9}=\frac{3}{2}.$$

从 x_1 出发, 沿 d_1 搜索:

$$x_1+\alpha d_1=\begin{bmatrix}2/3\\4/3\\1\\0\end{bmatrix}+\alpha\begin{bmatrix}-4/9\\4/9\\4/3\\0\end{bmatrix}=\frac{1}{9}\begin{bmatrix}6-4\alpha\\12+4\alpha\\9+8\alpha\\0\end{bmatrix},$$

$$f(x_1+\alpha d_1)=\frac{32}{81}\alpha^2-\frac{24}{81}\alpha-\frac{117}{81}.$$

求解一维极小问题

$$\begin{aligned}&\min \frac{32}{81}\alpha^2-\frac{24}{81}\alpha-\frac{117}{81},\\&\text{s.t. } 0\le\alpha\le\frac{3}{2},\end{aligned}$$

得 $\alpha_1 = \frac{3}{8}$, 从而 $x_2 = x_1 + \alpha_1 d_1 = \left(\frac{1}{2}, \frac{3}{2}, \frac{4}{3}, 0\right)^T$.

第 3 次迭代. $J_2 = \{2, 3\}$. $\nabla f(x_2) = (-1, -1, 0, 0)^T$.

$$x_2^B = \begin{bmatrix} \frac{3}{2} \\ \frac{4}{3} \end{bmatrix}, \quad x_2^N = \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix}, \quad B = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}, \quad N = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}.$$

计算

$$B^{-1}N = \begin{bmatrix} 1 & 1 \\ 3 & 1 \end{bmatrix}, \quad \nabla_N f(x_1) = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad \nabla_B f(x_1) = \begin{bmatrix} -1 \\ 0 \end{bmatrix}.$$

计算简约梯度 $r(x_1^N)$ 得

$$\begin{aligned} r(x_1^N) &= \nabla_N f(x_1) - (B^{-1}N)^T \nabla_B f(x_1) \\ &= \begin{bmatrix} -1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 3 & 1 \end{bmatrix}^T \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned}$$

当 $r_j(x_1^N) > 0$ 时, 取 $(d_1^N)_j = -(x_1^N)_j r_j(x_1^N)$, 而 $r_j(x_1^N) = 0$ 时, 取 $(d_1^N)_j = -r_j(x_1^N)$. 故

$$d_2^N = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad d_2^B = -(B^{-1}N)d_2^N = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

从而 $d_2 = (0, 0, 0, 0)^T$. 根据定理 53, x_2 即为 KT 点, 故 $x^* = \left(\frac{1}{2}, \frac{3}{2}\right)^T$ 是原问题的全局极小点. □

10.3.2　广义简约梯度法

Abadie 和 Carpentier 于 1969 年将 Wolfe 简约梯度法推广到一般非线性约束的情形, 提出了所谓的广义简约梯度法. 设一般非线性约束优化问题为

$$\begin{cases} \min f(x), \\ \text{s.t. } h_i(x) = 0, \; i \in E = \{1, \cdots, l\}, \\ \qquad g_i(x) \geq 0, \; i \in I = \{1, \cdots, m\}, \end{cases}$$

其中 f, $h_i\,(i \in E)$, $g_i\,(i \in I)$ 是连续可微的函数.

假设 x_k 是第 k 次可行迭代点, 记 $I_k = E \cup \{i \,|\, i \in I,\ g_i(x_k) = 0\}$,并设

$$c(x_k) = \big(h_1(x_k), \cdots, h_l(x_k); g_i(x_k)\,(i \in I_k \backslash E)\big)^T.$$

则在第 k 步可考虑子问题:

$$\begin{cases} \min f(x), \\ \text{s.t. } c(x_k) = 0. \end{cases} \tag{10.49}$$

现在我们讨论如何确立简约梯度 $r(x_k^N)$. 不失一般性, 设 $I_k = \{1, 2, \cdots, s\}$, $s \ge l$, 且约束函数的 Jacobi 矩阵

$$\Big[\nabla h_1(x_k), \cdots, \nabla h_l(x_k),\ \nabla g_i(x_k)\,(i \in I_k \backslash E)\Big]^T$$

行满秩 (不妨设其前 s 列构成的方阵非奇异). 那么可以解出前 s 个变量, 即可以用其余的 $n-s$ 个变量来表示这 s 个变量. 通常称这 s 个变量组成的子向量为基向量, 记为 x_k^B, 其余 $n-s$ 个变量组成的子向量为非基向量, 记为 x_k^N. 为方便计, 去掉下标 k, 并记 $s \times s$ 矩阵

$$\nabla_B c(x) = \Big[\nabla_B h_1(x), \cdots, \nabla_B h_l(x), \nabla_B g_1(x), \cdots,\ \nabla_B g_{s-l}(x)\Big]^T,$$

其中

$$c(x) = \big(h_1(x), \cdots, h_l(x), g_1(x), \cdots, g_{s-l}(x)\big)^T := \big(c_1(x), \cdots, c_s(x)\big)^T$$

及

$$\nabla_B c_i(x) = \begin{bmatrix} \dfrac{\partial c_i(x)}{\partial x_1} \\ \vdots \\ \dfrac{\partial c_i(x)}{\partial x_s} \end{bmatrix}, \ i = 1, 2, \cdots, s.$$

由假设可知 $\nabla_B c(x)$ 非奇异. 再记矩阵

$$\nabla_N c(x) = \Big[\nabla_N c_1(x),\ \nabla_N c_2(x),\ \cdots,\ \nabla_N c_s(x)\Big]^T \in \mathbb{R}^{s\times(n-s)},$$

其中

$$\nabla_N c_i(x) = \begin{bmatrix} \dfrac{\partial c_i(x)}{\partial x_{s+1}} \\ \vdots \\ \dfrac{\partial c_i(x)}{\partial x_n} \end{bmatrix}, \ i = 1, 2, \cdots, s.$$

在 x 的某邻域内, 由非线性方程组 $c(x)=c(x_B,x_N)=0$ 可以确定 x_B 为 x_N 的函数, 即 $x_B=\varphi(x_N)$.

对等式 $c(x)=c(x_B,x_N)=0$ 两边关于 x_N 求梯度得 $J_{BN}(x_N)\nabla_B c(x)+\nabla_N c(x)=0$, 其中

$$J_{BN}(x_N)=\left[\frac{\partial(x_1,\cdots,x_s)}{\partial(x_{s+1},\cdots,x_n)}\right]^T=\begin{bmatrix}\dfrac{\partial x_1}{\partial x_{s+1}} & \dfrac{\partial x_2}{\partial x_{s+1}} & \cdots & \dfrac{\partial x_s}{\partial x_{s+1}}\\ \vdots & \vdots & & \vdots\\ \dfrac{\partial x_1}{\partial x_n} & \dfrac{\partial x_2}{\partial x_n} & \cdots & \dfrac{\partial x_s}{\partial x_n}\end{bmatrix}.$$

从而, 我们有

$$J_{BN}(x_N)=-\nabla_N c(x)\big[\nabla_B c(x)\big]^{-1}. \tag{10.50}$$

注意到 $f(x)\equiv f(\varphi(x_N),x_N)$, 对其求关于 x_N 的梯度 (即简约梯度) 得

$$r(x_N)=\nabla_N f(x)+J_{BN}(x_N)\nabla_B f(x).$$

将 (10.50) 代入上式即得

$$r(x_N)=\nabla_N f(x)-\nabla_N c(x)\big[\nabla_B c(x)\big]^{-1}\nabla_B f(x). \tag{10.51}$$

现在设下降可行方向为 $d=(d_B^T,d_N^T)^T$, 则由下降可行条件知 d 应满足

$$\nabla f(x)^T d<0,\qquad \nabla c(x)^T d=0.$$

由 $\nabla c(x)^T d=0$ 可得

$$\nabla_B c(x)^T d_B+\nabla_N c(x)^T d_N=0.$$

于是有

$$d_B=-[\nabla_B c(x)^T]^{-1}\nabla_N c(x)^T d_N=J_{BN}(x_N)^T d_N. \tag{10.52}$$

又由 $\nabla f(x)^T d<0$, 得

$$\nabla_B f(x)^T d_B+\nabla_N f(x)^T d_N<0.$$

将 d_B 的表达式代入上式得

$$\nabla_N f(x)^T d_N-\nabla_B f(x)^T[\nabla_B c(x)^T]^{-1}\nabla_N c(x)^T d_N<0,$$

即

$$r(x_N)^T d_N < 0.$$

因此, d_N 的一种简单的选取方法是 $d_N = -r(x_N)$. 至此, 我们在上述的推导过程中以 $x = x_k$ 代入得 x_k 处的简约梯度为 $r(x_k^N)$, 下降可行方向为

$$d_k = \begin{bmatrix} d_k^B \\ d_k^N \end{bmatrix} = \begin{bmatrix} -J_{BN}(x_k^N)^T r(x_k^N) \\ -r(x_k^N) \end{bmatrix} = \begin{bmatrix} -J_{BN}(x_k^N)^T \\ -I_{n-s} \end{bmatrix} r(x_k^N) \tag{10.53}$$

最后确定步长 α_k. 可以通过求解下述一维极小问题

$$\begin{cases} \min f(x_k + \alpha d_k), \\ \text{s.t. } c_i(x_k + \alpha d_k) = 0,\ i \in E, \\ \qquad c_i(x_k + \alpha d_k) \geq 0,\ i \in I \end{cases} \tag{10.54}$$

获得搜索步长 α_k, 然后令 $x_{k+1} := x_k + \alpha_k d_k$ 即得到后继可行迭代点 x_{k+1}.

最后讨论拉格朗日乘子的估计. 由定理 39 知, 在极小点 x^* 处成立

$$\nabla f(x^*) = \sum_{i \in E} \mu_i^* \nabla h_i(x^*) + \sum_{i \in I^*} \lambda_i^* \nabla g_i(x^*), \tag{10.55}$$

其中 $\lambda_i^* \geq 0$, $i \in I^* = \{i \,|\, g_i(x^*) = 0\}$. 记

$$\begin{aligned} &\nabla c(x^*) = \big[\nabla h_1(x^*), \cdots, \nabla h_l(x^*),\ \nabla g_i(x^*),\ (i \in I^*)\big], \\ &\nu^* = \big((\mu^*)^T,\ (\lambda^*)^T\big)^T = \big(\mu_1, \cdots, \mu_l, \lambda_i\,(i \in I^*)\big)^T. \end{aligned}$$

那么 (10.55) 可以写成 $\nabla f(x^*) = \nabla c(x^*)\nu^*$. 由广义逆知识可得其极小最小二乘解 $\nu^* = [\nabla c(x^*)]^+ \nabla f(x^*)$. 因此, 计算相应的乘子估计

$$\nu_k = (\mu_k^T, \lambda_k^T)^T = [\nabla c(x_k)]^+ \nabla f(x_k). \tag{10.56}$$

下面给出广义简约梯度法的详细计算步骤.

算法 29 (**广义简约梯度法**)

步 0 选取初始值. 给定初始可行点 $x_0 \in \mathbb{R}^N$, $0 \leq \varepsilon \ll 1$. 令 $k := 0$.

步 1 检验终止条件. 确定基变量 x_k^B 和非基变量 x_k^N. 由 (10.51) 计算简约梯度 $r(x_k^N)$. 若 $\|r(x_k^N)\| \leq \varepsilon$, 则 x_k 为近似极小点, 停算.

步 2 确定搜索方向. 由 (10.53) 计算下降可行方向 d_k.

步 3 进行线搜索. 解子问题 (10.54) 得步长因子 α_k. 令 $x_{k+1} := x_k + \alpha_k d_k$.

步 4 修正有效集. 先求 x_{k+1} 处的有效集, 设为 $\bar{I}_{k+1}$. 由 (10.56) 计算 λ_k. 若 $\lambda_{k+1} \geq 0$, 则 $I_{k+1} = \bar{I}_{k+1}$. 否则, I_{k+1} 是 $\bar{I}_{k+1}$ 中删去 λ_{k+1} 最小分量所对应的约束指标集.

步 5 令 $k := k+1$, 转步 1.

注 (1) 在算法 29 的步 2 中, 当 $\|r(x_k^N)\| \leq \varepsilon$ 时, 实际还需要判别对应于不等式约束的拉格朗日乘子的非负性, 若不满足还需进行改进. (2) 广义简约梯度法通过消去某些变量在降维空间中运算, 能够较快确定最优解, 可用来求解大型问题, 因而它是目前求解非线性优化问题的最有效的方法之一.

习 题 10

1. 设有下列最优化问题

$$
\begin{aligned}
\min\quad & f(x) = x_1^2 + x_1x_2 + 2x_2^2 - 6x_1 - 2x_2 - 12x_3,\\
\text{s.t.}\quad & x_1 + x_2 + x_3 = 2,\\
& x_1 - 2x_2 \geq -3,\\
& x_1, x_2, x_3 \geq 0.
\end{aligned}
$$

求出在点 $\bar{x} = (1,1,0)^T$ 处的一个下降可行方向.

2. 用Zoutendijk 方法求解下列问题:

$$
\begin{aligned}
(1)\quad \min\quad & f(x) = x_1^2 + 4x_2^2 - 34x_1 - 32x_2,\\
\text{s.t.}\quad & -2x_1 - x_2 + 6 \geq 0,\\
& -x_2 + 2 \geq 0,\\
& x_1 \geq 0, x_2 \geq 0.
\end{aligned}
$$

取初始点为 $x_0 = (1,2)^T$.

$$
\begin{aligned}
(2)\quad \min\quad & f(x) = x_1^2 + x_2^2 - x_1x_2 - 2x_1 + 3x_2,\\
\text{s.t.}\quad & -x_1 - x_2 + 3 \geq 0,\\
& -x_1 - 5x_2 + 6 \geq 0,\\
, \quad & x_1, x_2 \geq 0.
\end{aligned}
$$

取初始点为 $x_0 = (2,0)^T$.

3. 用梯度投影法求解下列优化问题:

$$
\begin{aligned}
(1)\quad \min\quad & f(x)=x_1^2+2x_2^2+x_1x_2-6x_1-2x_2-12x_3,\\
\text{s.t.}\quad & x_1+x_2+x_3-2=0,\\
& x_1-2x_2+3\ge 0,\\
& x_1,x_2,x_3\ge 0.
\end{aligned}
$$

取初始点为 $x_0=(1,0,1)^T$.

$$
\begin{aligned}
(2)\quad \min\quad & f(x)=(4-x_2)(x_1-3)^2,\\
\text{s.t.}\quad & -x_1-x_2+3\ge 0,\\
& -x_1+2\ge 0,\\
& -x_2+2\ge 0,\\
& x_1,x_2\ge 0.
\end{aligned}
$$

取初始点为 $x_0=(1,2)^T$.

4. 用简约梯度法求解下列问题:

$$
\begin{aligned}
(1)\quad \min\quad & f(x)=(x_1-2)^2+(x_2-2)^2,\\
\text{s.t.}\quad & -x_1-x_2+2\ge 0,\\
& x_1,x_2\ge 0
\end{aligned}
$$

$$
\begin{aligned}
(2)\quad \min\quad & f(x)=x_1^2+x_2^2-2x_1-4x_2,\\
\text{s.t.}\quad & -x_1-x_2+1\ge 0,\\
& x_1,x_2\ge 0
\end{aligned}
$$

5. 设 (z_k,d_k) 是下列线性规划

$$
\begin{aligned}
\min\quad & z,\\
\text{s.t.}\quad & \nabla f(x_k)^Td-z\le 0,\\
& \nabla g_i(x_k)^Td+z\ge 0,\ i\in I(x_k),\\
& -1\le d_i\le 1
\end{aligned}
$$

的最优解且 $z_k<0$, 这里 $I(x_k)=\{i|g_i(x_k)=0\}$. 试证明 d_k 是下列问题

$$
\begin{aligned}
\min\quad & f(x),\\
\text{s.t.}\quad & g_i(x)\ge 0, i=1,\cdots,m
\end{aligned}
$$

在 x_k 处的一个下降可行方向.

6. 证明关于投影矩阵 P 的如下性质:

(1) 若 $P\in\mathbb{R}^{n\times n}$ 是投影矩阵，则 P 是半正定的;

(2) 矩阵 P 是投影矩阵的充要条件是 $Q = I - P$ 是投影矩阵.

(3) 设 P 是投影矩阵, 则

$$L := \{Px | x \in \mathbb{R}^n\} \text{ 与 } L^{\perp} := \{(I-P)x | x \in \mathbb{R}^n\}$$

相互正交, 而且 $\forall x \in \mathbb{R}^n$ 可唯一分解为 $x = y + z$, $y \in L$, $z \in L^{\perp}$.

7. 对任一 $x \in \mathbb{R}^n$ 在子空间 L 上的正交投影 y, 证明 $\|x - y\|_2^2 = \min_{z \in L} \|x - z\|_2^2$.

8. 设 $X \subset \mathbb{R}^n$ 是闭凸集, 证明下面的不等式:

(1) $[P_X(x) - x]^T[y - P_X(x)] \geq 0$, $\forall x \in \mathbb{R}^n$, $y \in X$;

(2) $[P_X(x) - P_X(y)]^T(x - y) \geq \|P_X(x) - P_X(y)\|^2$, $\forall x, y \in \mathbb{R}^n$.

9. 用广义简约梯度法解下列优化问题

$$\begin{aligned}
\min \quad & f(x) = x_1^2 + 2x_1x_2 + x_2^2 + 12x_1 - 4x_2 + 3, \\
\text{s.t.} \quad & x_1 - x_2 = 0, \\
& 1 \leq x_1 \leq 3, \\
& 1 \leq x_2 \leq 3.
\end{aligned}$$

第十一章　二次规划

二次规划是非线性优化中的一种特殊情形, 它的目标函数是二次实函数, 约束函数都是线性函数. 由于二次规划比较简单, 便于求解 (仅次于线性规划), 并且一些非线性优化问题可以转化为求解一系列的二次规划问题 (即本书第十二章所介绍的“序列二次规划法”), 因此二次规划的求解方法较早引起人们的重视, 成为求解非线性优化的一个重要途径. 二次规划的算法较多, 本章仅介绍求解等式约束凸二次规划的零空间方法和拉格朗日方法以及求解一般约束凸二次规划的有效集方法.

11.1　等式约束凸二次规划的解法

我们考虑如下的二次规划问题

$$\begin{cases} \min\ \dfrac{1}{2}x^T Hx + c^T x, \\ \text{s.t.}\ \ Ax = b, \end{cases} \tag{11.1}$$

其中 $H \in \mathbb{R}^{n\times n}$ 对称正定, $A \in \mathbb{R}^{m\times n}$ 行满秩, $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$. 本节我们介绍两种求解问题 (11.1) 的数值方法, 即零空间方法和值空间方法（通常称为拉格朗日方法）.

11.1.1　零空间方法

设 x_0 满足 $Ax_0 = b$. 记 A 的零空间为

$$\mathcal{N}(A) = \big\{z \in \mathbb{R}^n \,|\, Az = 0\big\},$$

则问题 (11.1) 的任一可行点 x 可表示成 $x = x_0 + z$, $z \in \mathcal{N}(A)$. 这样, 问题 (11.1) 可等价变形为

$$\begin{cases} \min \ \dfrac{1}{2}z^T Hz + z^T(c + Hx_0), \\ \text{s.t.} \ \ Az = 0. \end{cases} \tag{11.2}$$

令 $Z \in \mathbb{R}^{n\times(n-m)}$ 是 $\mathcal{N}(A)$ 的一组基组成的矩阵, 那么, 对任意的 $d \in \mathbb{R}^{n-m}$, 有 $z = Zd \in \mathcal{N}(A)$. 于是问题 (11.2) 变为无约束优化问题

$$\min \frac{1}{2}d^T(Z^T HZ)d + d^T[Z^T(c + Hx_0)]. \tag{11.3}$$

容易发现, 当 H 半正定时, $Z^T HZ$ 也是半正定的. 此时, 若 d^* 是 (11.3) 的稳定点, d^* 也是 (11.3) 的全局极小点, 同时 $x^* = x_0 + Zd^*$ 是 (11.1) 的全局极小点, $\lambda^* = A^+(Hx^* + c)$ 是相应的拉格朗日乘子, 其中 A^+ 是矩阵 A 的 Penrose 广义逆. 由于这种方法是基于约束函数的系数矩阵的零空间, 因此把它称之为零空间方法.

余下的问题就是如何确定可行点 x_0 和零空间 $\mathcal{N}(A)$ 的基矩阵 Z. 有多种方法来确定这样的 x_0 和 Z. 我们在此介绍 1974 年 Gill 和 Murry 所提出的一种方法, 即先对 A^T 作 QR 分解

$$A^T = Q\begin{bmatrix} R \\ 0 \end{bmatrix} = \begin{bmatrix} Q_1, \ Q_2 \end{bmatrix}\begin{bmatrix} R \\ 0 \end{bmatrix}, \tag{11.4}$$

其中, Q 是一个 n 阶正交阵, R 是一个 m 上三角阵, $Q_1 \in \mathbb{R}^{n\times m}$, $Q_2 \in \mathbb{R}^{n\times(n-m)}$. 那么确立 x_0 和 Z 为

$$x_0 = Q_1 R^{-T} b, \quad Z = Q_2, \tag{11.5}$$

同时有

$$A^+ = Q_1 R^{-T}. \tag{11.6}$$

下面写出零空间方法的算法步骤:

算法 30 (零空间方法)

步 0　数据准备. 确定矩阵 H, A 和向量 c, b.

步 1　由 (11.4) 对 A^T 进行 QR 分解, 得 Q_1, Q_2 和 R.

步 2　按 (11.5) 计算可行点 x_0 和零空间 $\mathcal{N}(A)$ 的基矩阵 Z。

步 3　求解无约束优化子问题 (11.3) 得解 d^*.

步 4 计算全局极小点 $x^* = x_0 + Zd^*$ 和相应的拉格朗日乘子 $\lambda^* = A^+(Hx^* + c)$, 其中 A^+ 由 (11.6) 确定.

11.1.2 拉格朗日方法及其 Matlab 程序

下面我们来推导用拉格朗日乘子法解问题 (11.1) 的求解公式.

首先写出拉格朗日函数:

$$L(x, \lambda) = \frac{1}{2}x^T Hx + c^T x - \lambda^T (Ax - b), \tag{11.7}$$

令

$$\nabla_x L(x, \lambda) = 0, \quad \nabla_\lambda L(x, \lambda) = 0,$$

得到方程组

$$\begin{aligned} Hx - A^T\lambda &= -c, \\ -Ax &= -b. \end{aligned}$$

将上述方程组写成分块矩阵形式:

$$\begin{bmatrix} H & -A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -c \\ -b \end{bmatrix}. \tag{11.8}$$

我们称上述方程组的系数矩阵

$$\begin{bmatrix} H & -A^T \\ -A & 0 \end{bmatrix}$$

为拉格朗日矩阵.

下面的定理给出了线性方程组 (11.8) 有唯一解的充分条件.

定理 54 设 $H \in \mathbb{R}^{n\times n}$ 对称正定, $A \in \mathbb{R}^{m\times n}$ 行满秩. 若在问题 (11.1) 的解 x^* 处满足二阶充分条件, 即

$$d^T Hd > 0, \ \forall\, d \in \mathbb{R}^n, \ d \neq 0, \ Ad = 0,$$

则线性方程组 (11.9) 的系数矩阵非奇异, 即方程组 (11.9) 有唯一解.

证 设 (d,ν) 是下面的齐次线性方程组的解:

$$\begin{bmatrix} H & -A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} d \\ \nu \end{bmatrix} = 0, \tag{11.9}$$

即

$$Hd - A^T\nu = 0, \quad Ad = 0.$$

故

$$d^T Hd = d^T A^T \nu = 0, \quad Ad = 0.$$

于是由二阶充分性条件必有 $d=0$. 从而

$$A^T\nu = Hd = 0.$$

注意到 A 行满秩, 故必有 $\nu = 0$. 由此可知, 齐次线性方程组 (11.9) 只有零解, 因此其系数矩阵必然非奇异. 证毕. □

下面我们来导出方程 (11.8) 的求解公式. 根据定理 54, 拉格朗日矩阵必然是非奇异的, 故可设其逆为

$$\begin{bmatrix} H & -A^T \\ -A & 0 \end{bmatrix}^{-1} = \begin{bmatrix} G & -B^T \\ -B & C \end{bmatrix}.$$

由恒等式

$$\begin{bmatrix} H & -A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} G & -B^T \\ -B & C \end{bmatrix} = \begin{bmatrix} I_n & 0_{n\times m} \\ 0_{m\times n} & I_m \end{bmatrix}$$

可得

$$HG + A^T B = I_n, \quad -HB^T - A^T C = 0_{n\times m},$$

$$-AG = 0_{m\times n}, \qquad AB^T = I_m.$$

于是由上述 4 个等式得到矩阵 G, B, C 的表达式

$$G = H^{-1} - H^{-1}A^T(AH^{-1}A^T)^{-1}AH^{-1}, \tag{11.10}$$

$$B = (AH^{-1}A^T)^{-1}AH^{-1}, \tag{11.11}$$

$$C = -(AH^{-1}A^T)^{-1}. \tag{11.12}$$

因此, 由 (11.8) 可得解的表达式

$$\begin{bmatrix} \bar{x} \\ \bar{\lambda} \end{bmatrix} = \begin{bmatrix} G & -B^T \\ -B & C \end{bmatrix} \begin{bmatrix} -c \\ -b \end{bmatrix} = \begin{bmatrix} -Gc + B^T b \\ Bc - Cb \end{bmatrix}, \tag{11.13}$$

其中 G, B, C 分别由 (11.10), (11.11) 和 (11.12) 给出.

下面给出 $\bar{x}$ 和 $\bar{\lambda}$ 的另一种等价表达式. 设 x_k 是问题 (11.1) 的任一可行点, 即 x_k 满足 $Ax_k = b$. 而在此点处目标函数的梯度为 $g_k = \nabla f(x_k) = Hx_k + c$. 利用 x_k 和 g_k, 可将 (11.13) 改写为

$$\begin{bmatrix} \bar{x} \\ \bar{\lambda} \end{bmatrix} = \begin{bmatrix} x_k - Gg_k \\ Bg_k \end{bmatrix}. \tag{11.14}$$

下面我们给出求解等式约束二次规划拉格朗日方法的 Matlab 程序.

程序 16 本程序用拉格朗日方法求解等式约束条件的二次规划问题.

```
function [x,lam,fval]=qlag(H,A,b,c)
% 功能: 用拉格朗日方法求解等式约束二次规划:
%            min f(x)=0.5*x'Hx+c'x, s.t. Ax=b
%输入: H,c分别是目标函数的矩阵和向量, A,b分别是
%           约束条件中的矩阵和向量
%输出: (x, lam) 是 KT 点, fval 是最优值.
IH=inv(H);
AHA=A*IH*A';
IAHA=inv(AHA);
AIH=A*IH;
G=IH-AIH'*IAHA*AIH;
B=IAHA*AIH;
C=-IAHA;
x=B'*b-G*c;
lam=B*c-C*b;
fval=0.5*x'*H*x+c'*x;
```

我们利用上述程序求解一个二次规划问题.

例 32 利用程序 16 求解下列问题

$$\min x_1^2+2x_2^2+x_3^2-2x_1x_2+x_3,$$
$$\text{s.t.}\ \ x_1+x_2+x_3=4,$$
$$2x_1-x_2+x_3=2.$$

解 容易写出

$$H=\begin{bmatrix}2 & -2 & 0\\ -2 & 4 & 0\\ 0 & 0 & 2\end{bmatrix},\ c=\begin{bmatrix}0\\0\\1\end{bmatrix},\ A=\begin{bmatrix}1 & 1 & 1\\ 2 & -1 & 1\end{bmatrix},\ b=\begin{bmatrix}4\\2\end{bmatrix}.$$

在 Matlab 命令窗口依次输入:

```
H=[2 -2 0;-2 4 0; 0 0 2];
c=[0 0 1]';
A=[1 1 1;2 -1 1];
b=[4 2]';
[x,lam]=qlag(H,A,b,c)
```

得到

```
x =
    1.9091
    1.9545
    0.1364
lam =
     2.6364
    -1.3636
fval =
     3.9773
```

11.2　一般凸二次规划的有效集方法

考虑一般二次规划

$$\begin{cases}\min\ \dfrac{1}{2}x^THx+c^Tx,\\ \text{s.t.}\ \ a_i^Tx-b_i=0,\ i\in E=\{1,\cdots,l\},\\ \qquad a_i^Tx-b_i\ge 0,\ i\in I=\{l+1,\cdots,m\},\end{cases}\tag{11.15}$$

其中 H 是 n 阶对称阵. 记 $I(x^*) = \{i \mid a_i^T x^* - b_i = 0,\ i \in I\}$, 下面的定理给出了问题 (11.15) 的一个最优性充要条件, 其证明可参见文献 [1].

定理 55 x^* 是二次规划问题 (11.15) 的局部极小点当且仅当

(1) 存在 $\lambda^* \in \mathbb{R}^m$, 使得

$$\begin{cases} Hx^* + c - \sum\limits_{i\in E} \lambda_i^* a_i - \sum\limits_{i\in I} \lambda_i^* a_i = 0, \\ a_i^T x^* - b_i = 0,\ i \in E, \\ a_i^T x^* - b_i \ge 0,\ i \in I, \\ \lambda_i^* \ge 0,\ i \in I;\ \lambda_i^* = 0,\ i \in I \backslash I(x^*). \end{cases}$$

(2) 记

$$\mathcal{S} = \{d \in \mathbb{R}^n \backslash \{0\} \mid d^T a_i = 0, i \in E;\ d^T a_i \ge 0, i \in I(x^*);\ d^T a_i = 0, i \in I(x^*) \text{ 且 } \lambda_i^* > 0\}.$$

则对于任意的 $d \in \mathcal{S}$, 均有 $d^T H d \ge 0$.

容易发现, 问题 (11.15) 是凸二次规划的充要条件是 H 半正定. 此时, 定理 55 的第二部分自然满足. 注意到凸优化问题的局部极小点也是全局极小点的性质, 我们有下面的定理:

定理 56 x^* 是凸二次规划的全局极小点的充要条件是 x^* 满足 KT 条件, 即存在 $\lambda^* \in \mathbb{R}^m$, 使得

$$\begin{cases} Hx^* + c - \sum\limits_{i\in E} \lambda_i^* a_i - \sum\limits_{i\in I} \lambda_i^* a_i = 0, \\ a_i^T x^* - b_i = 0,\ i \in E, \\ a_i^T x^* - b_i \ge 0,\ i \in I, \\ \lambda_i^* \ge 0,\ i \in I;\ \lambda_i^* = 0,\ i \in I \backslash I(x^*). \end{cases}$$

下面我们介绍求解一般凸二次规划问题的有效集方法及其 Matlab 实现.

11.2.1　有效集方法的理论推导

首先引入下面的定理, 它是有效集方法理论基础, 其证明可参见文献 [2].

定理 57 设 x^* 是一般凸二次规划问题 (11.15) 的全局极小点, 且在 x^* 处的有效集为 $S(x^*) = E \cup I(x^*)$, 则 x^* 也是下列等式约束凸二次规划

$$\begin{cases} \min \ \dfrac{1}{2}x^T Hx + c^T x, \\ \text{s.t.} \ a_i^T x - b_i = 0, \ i \in S(x^*) \end{cases} \tag{11.16}$$

的全局极小点.

从上述定理可以发现, 有效集方法的最大难点是事先一般不知道有效集 $S(x^*)$, 因此只有想办法构造一个集合序列去逼近它. 即, 从初始点 x_0 出发, 计算有效集 $S(x_0)$, 解对应的等式约束子问题. 重复这一做法, 得到有效集序列 $\{S(x_k)\}$, $k = 0, 1, \cdots$, 使之 $S(x_k) \to S(x^*)$, 以获得原问题的最优解.

基于上述定理, 我们分 4 步来介绍有效集方法的算法原理和实施步骤.

第 1 步. 形成子问题并求出搜索方向 d_k. 设 x_k 是问题 (11.15) 的一个可行点, 据此确定相应的有效集 $S_k = E \cup I(x_k)$, 其中 $I(x_k) = \{i | a_i^T x_k - b_i = 0, i \in I\}$. 求解相应的子问题

$$\begin{cases} \min \ \dfrac{1}{2}x^T Hx + c^T x, \\ \text{s.t.} \ a_i^T x - b_i = 0, \ i \in S_k. \end{cases} \tag{11.17}$$

上述问题等价于

$$\begin{cases} \min \ q_k(d) = \dfrac{1}{2}d^T Hd + g_k^T d, \\ \text{s.t.} \ a_i^T d = 0, \ i \in S_k, \end{cases} \tag{11.18}$$

其中 $x = x_k + d$, $g_k = Gx_k + c$. 设求出问题 (11.18) 的全局极小点为 d_k, λ_k 是对应的拉格朗日乘子.

第 2 步. 进行线搜索确定步长因子 α_k. 假设 $d_k \neq 0$, 分两种情形讨论.

(1) 若 $x_k + d_k$ 是问题 (11.15) 的可行点, 即

$$a_i^T (x_k + d_k) - b_i = 0, \ i \in E \text{ 及 } a_i^T (x_k + d_k) - b_i \geq 0, \ i \in I.$$

则令 $\alpha_k = 1$, $x_{k+1} = x_k + d_k$.

(2) 若 $x_k + d_k$ 不是问题 (11.15) 的可行点, 则通过线搜索求出下降最好的可行点. 注意到目标函数是凸二次函数, 那么这一点应该在可行域的边界上达到. 因此只要求出满足可行条件的最大步长 α_k 即可.

当 $i \in S_k$ 时, 对于任意的 $\alpha_k \geq 0$, 都有 $a_i^T d_k = 0$ 和 $a_i^T(x_k+\alpha_k d_k) = a_i^T x_k = b_i$, 此时, $\alpha_k \geq 0$ 不受限制. 当 $i \notin S_k$ 时, 即第 i 个约束是严格的不等式约束, 此时要求 α_k 满足 $a_i^T(x_k + \alpha_k d_k) \geq b_i$, 即

$$\alpha_k a_i^T d_k \geq b_i - a_i^T x_k, \quad i \notin S_k.$$

注意到上式右端非正, 故当 $a_i^T d_k \geq 0$ 时, 上式恒成立. 而当 $a_i^T d_k < 0$ 时, 由上式可解得

$$\alpha_k \leq \frac{b_i - a_i^T x_k}{a_i^T d_k}.$$

故有

$$\alpha_k = \bar{\alpha}_k = \min\left\{\frac{b_i - a_i^T x_k}{a_i^T d_k} \,\middle|\, a_i^T d_k < 0\right\}.$$

合并 (1) 和 (2) 可得

$$\alpha_k = \min\{1, \bar{\alpha}_k\}. \tag{11.19}$$

第 3 步. 修正 S_k. 当 $\alpha_k = 1$, 有效集不变, 即 $S_{k+1} := S_k$. 而当 $\alpha_k < 1$ 时,

$$\alpha_k = \bar{\alpha}_k = \frac{b_{i_k} - a_{i_k}^T x_k}{a_{i_k}^T d_k},$$

故 $a_{i_k}^T(x_k+\alpha_k d_k) = b_{i_k}$, 因此在 x_{k+1} 处增加了一个有效约束, 即 $S_{k+1} := S_k \cup \{i_k\}$.

第 4 步. 考虑 $d_k = 0$ 的情形. 此时 x_k 是问题 (11.17) 的全局极小点. 若这时对应的不等式约束的拉格朗日乘子均为非负, 则 x_k 也是问题 (11.15) 的全局极小点, 迭代终止. 否则, 如果对应的不等式约束的拉格朗日乘子有负的分量, 那么需要重新寻找一个下降可行方向.

设 $\lambda_{j_k} < 0$, $j_k \in I(x_k)$. 现在要求一个下降可行方向 d_k, 满足 $g_k^T d_k < 0$ 且 $a_j^T d_k = 0$, $\forall j \in E$; $a_j^T d_k \geq 0$, $\forall j \in I(x_k)$. 为简便计, 按下述方式选取 d_k:

$$\begin{aligned} & a_{j_k}^T(x_k + d_k) > b_{j_k}, \\ & a_j^T(x_k + d_k) = b_j, \ \forall j \in S_k, j \neq j_k, \end{aligned}$$

即

$$\begin{cases} a_{j_k}^T d_k > 0, \\ a_j^T d_k = 0, \ \forall j \in S_k, j \neq j_k, \end{cases} \tag{11.20}$$

另一方面, 注意到 x_k 是子问题 (11.17) 的全局极小点, 故有

$$Hx_k + c - \sum_{i\in S_k} \lambda_i^k a_i = 0,$$

即

$$g_k = A_k \lambda_k,$$

其中

$$A_k = (a_i)_{i\in S_k}, \quad \lambda_k = (\lambda_i^k)_{i\in S_k}.$$

从而, $g_k^T d_k = \lambda_k^T A_k^T d_k$. 由 (11.20) 知

$$A_k^T d_k = \sum_{j\in S_k} (a_j^T d_k)\, e_j = (a_{j_k}^T d_k)\, e_{j_k},$$

于是有

$$g_k^T d_k = \lambda_k^T (a_{j_k}^T d_k)\, e_{j_k} = \lambda_{j_k}^k (a_{j_k}^T d_k) < 0.$$

上式表明, 由 (11.20) 确定的 d_k 是一个下降可行方向. 因此, 令 $S_k' = S_k \backslash \{j_k\}$, 则修正后的子问题

$$\begin{cases} \min \ q_k(d) = \dfrac{1}{2} d^T H d + g_k^T d, \\ \text{s.t.} \ a_i^T d = 0, \ i \in S_k' \end{cases}$$

的全局极小点必然是原问题的一个下降可行方向.

11.2.2　有效集方法的算法步骤

经过上面的分析和推导, 我们现在可以写出有效集方法的算法步骤.

算法 31 (有效集方法)

步 0　选取初值. 给定初始可行点 $x_0 \in \mathbb{R}^n$, 令 $k := 0$.

步 1　解子问题. 确定相应的有效集 $S_k = E \cup I(x_k)$. 求解求解子问题

$$\begin{cases} \min \ q_k(d) = \dfrac{1}{2} d^T H d + g_k^T d, \\ \text{s.t.} \ a_i^T d = 0, \ i \in S_k, \end{cases}$$

得极小点 d_k 和拉格朗日乘子向量 λ_k. 若 $d_k \neq 0$ 转步 3; 否则, 转步 2.

步 2　检验终止准则. 计算拉格朗日乘子

$$\lambda_k = B_k g_k,$$

其中

$$g_k = Hx_k + c, \quad B_k = (A_k H^{-1} A_k^T)^{-1} A_k H^{-1}, \quad A_k = (a_i)_{i \in S_k}.$$

令

$$(\lambda_k)_t = \min_{i \in I(x_k)} \{(\lambda_k)_i\}.$$

若 $(\lambda_k)_t \geq 0$, 则 x_k 是全局极小点, 停算. 否则, 若 $(\lambda_k)_t < 0$, 则令 $S_k := S_k \backslash \{t\}$, 转步 1.

步 3　确定步长 α_k. 令 $\alpha_k = \min\{1, \bar{\alpha}_k\}$, 其中

$$\bar{\alpha}_k = \min_{i \notin S_k} \left\{ \frac{b_i - a_i^T x_k}{a_i^T d_k} \,\middle|\, a_i^T d_k < 0 \right\}.$$

令 $x_{k+1} := x_k + \alpha_k d_k$.

步 4　若 $\alpha_k = 1$, 则令 $S_{k+1} := S_k$; 否则, 若 $\alpha_k < 1$, 则令 $S_{k+1} = S_k \cup \{j_k\}$, 其中 j_k 满足

$$\bar{\alpha}_k = \frac{b_{j_k} - a_{j_k}^T x_k}{a_{j_k}^T d_k}.$$

步 5　令 $k := k + 1$, 转步 1.

下面给出算法 31 的收敛性定理.

定理 58　假设问题 (11.15) 中的矩阵 H 对称正定. 若在算法 31 每步迭代中的矩阵

$$A_k = (a_i)_{i \in S_k}$$

列满秩, 且 $\alpha_k \neq 0$, 则算法 31 在有限步之内得到问题 (11.15) 的全局极小点.

证　注意到, 若 $d_k = 0$, 则 x_k 是子问题 (11.17) 的 KT 点和全局极小点. 若 $d_k \neq 0$ 且 $\alpha_k = 1$, 则 $S_{k+1} = S_k$, 这时关于 x_{k+1} 的子问题仍为 (11.17), 所以, x_{k+1} 是 (11.17) 的全局极小点. 只有当 $\alpha_k < 1$ 时, x_{k+1} 才不是 (11.17) 的全局极小点, 但这时要加进一个约束, 形成新的子问题. 这样的过程最多连续 n 次, 因为这时子问题至少有 n 个等式约束, 从而 x_k 为唯一的可行点, 因而是对应的子问题 (11.17) 的全局极小点.

另一方面, 子问题 (11.17) 虽然约束条件在变化, 但其目标函数与原问题 (11.15) 是一致的. 由于 $\alpha_k \neq 0$, 故每次迭代目标函数值减少. 再注意到 H 正定, 因此子问题的全局极小点是唯一的, 从而不会出现子问题的全局极小点被两个不同的 x_k 达到的情形. 而约束个数的有限性保证了有效集 S_k 不同个数的有限性. 因此, 不失一般性, 设有效集 S_k 不同个数为 N_0, 那么不同子问题的个数也为 N_0. 故而, 迭代至多在 $N_0 n$ 步之后, x_k 遍历所有子问题的全局极小点. 由定理 57 知, 算法 31 在有限步之内达到问题 (11.15) 的全局极小点. □

例 33　用有效集方法求解下列二次规划问题

$$\begin{cases} \min\ \ f(x) = x_1^2 - x_1x_2 + 2x_2^2 - x_1 - 10x_2, \\ \text{s.t.}\ -3x_1 - 2x_2 \geq -6, \\ \qquad x_1 \geq 0,\ x_2 \geq 0. \end{cases}$$

解　首先确定矩阵 H 和向量 c:

$$H = \begin{bmatrix} 2 & -1 \\ -1 & 4 \end{bmatrix}, \quad c = \begin{bmatrix} -1 \\ -10 \end{bmatrix}.$$

取初始可行点 $x_0 = (0,0)^T$. 在 x_0, 有效集 $S_0 = \{2,3\}$. 求解相应的子问题

$$\begin{cases} \min\ \ d_1^2 - d_1d_2 + 2d_2^2 - d_1 - 10d_2, \\ \text{s.t.}\ \ d_1 = 0,\ \ d_2 = 0. \end{cases}$$

得解 $d_0 = (0,0)^T$. 因此, x_0 是相应的子问题 (11.17) 的最优解. 计算拉格朗日乘子. 由 $S_0 = \{2,3\}$ 知

$$A_0 = (a_i)_{i\in S_0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad g_0 = Hx_0 + c = \begin{bmatrix} -1 \\ -10 \end{bmatrix}.$$

故

$$\begin{bmatrix} (\lambda_0)_2 \\ (\lambda_0)_3 \end{bmatrix} = [(A_0H^{-1}A_0^T)^{-1}A_0H^{-1}]g_0 = \begin{bmatrix} -1 \\ -10 \end{bmatrix}.$$

由此可知, x_0 不是所求问题的最优解.

将 $(\lambda_0)_3=-10$ 对应的约束, 即原问题的第 3 个约束从有效集 S_0 中去掉, 置 $S_0=\{2\}$, 再解相应的子问题:

$$\begin{cases} \min\ d_1^2-d_1d_2+2d_2^2-d_1-10d_2, \\ \text{s.t.}\ \ d_1=0. \end{cases}$$

得解 $d_0=\left(0,\dfrac{5}{2}\right)^T$.

由于 $d_0\neq 0$, 需要计算步长 α_0. 注意到

$$\begin{aligned} \bar{\alpha} &= \min\left\{\frac{b_i-a_i^Tx_0}{a_i^Td_0}\middle| i\notin S_0, a_i^Td_0<0\right\} \\ &= \frac{-6}{(-3,-2)\cdot(0,\ 2.5)^T}=\frac{-6}{-5}=1.2. \end{aligned}$$

故 $\alpha_0=\min\{1,\bar{\alpha}\}=1$. 令

$$x_1=x_0+\alpha_0d_0=(0,0)^T+1\cdot\left(0,\frac{5}{2}\right)^T=\left(0,\frac{5}{2}\right)^T.$$

计算出 $g_1=\nabla f(x_1)=\left(-\dfrac{7}{2},0\right)^T$.

因 $\alpha_0=1$, 置 $S_1=\{2\}\,(=S_0)$. 在 x_1 处, 计算相应的拉格朗日乘子. 此时 $A_1=(1,0)$, 那么,

$$(\lambda_1)_2=[(A_1H^{-1}A_1^T)^{-1}A_1H^{-1}]g_1=-\frac{7}{2}.$$

由于 $(\lambda_1)_2<0$, 故 x_1 不是问题的最优解. 于是将指标 2 从 S_1 中剔除掉, 则更新后的 $S_1=\emptyset$. 再解相应的子问题

$$\min\ d_1^2-d_1d_2+2d_2^2-\frac{7}{2}d_1,$$

得解向量 $d_1=\left(2,\dfrac{1}{2}\right)^T$.

由于 $d_1\neq 0$, 需要计算步长 α_1. 注意到

$$\begin{aligned} \bar{\alpha} &= \min\left\{\frac{b_i-a_i^Tx_1}{a_i^Td_1}\middle| i\notin S_1, a_i^Td_1<0\right\} \\ &= \frac{-6-(-3,-2)\cdot(0,\ 2.5)^T}{(-3,-2)\cdot(2,\ 0.5)^T}=\frac{1}{7}. \end{aligned}$$

故
$$\alpha_1 = \min\{1, \bar{\alpha}\} = \frac{1}{7}.$$
令
$$x_2 = x_1 + \alpha_1 d_1 = \left(0,\ \frac{5}{2}\right)^T + \frac{1}{7}\left(2,\ \frac{1}{2}\right)^T = \left(\frac{2}{7},\ \frac{18}{7}\right)^T.$$
计算出 $g_2 = \nabla f(x_2) = (-3,\ 0)^T$.

在 x_2 处, 第一个约束是有效约束, 即 $S_2 = \{1\}$, 解相应的子问题:
$$\begin{cases} \min\ \ d_1^2 - d_1 d_2 + 2d_2^2 - 3d_1, \\ \text{s.t.}\ \ 3d_1 - 2d_2 = 0. \end{cases}$$
得解向量 $d_2 = \left(\frac{3}{14}, -\frac{9}{28}\right)^T$. 计算步长 α_2. 因
$$\begin{aligned} \bar{\alpha} &= \min\left\{\frac{b_i - a_i^T x_2}{a_i^T d_2}\Big| i \notin S_2, a_i^T d_2 < 0\right\} \\ &= \frac{0 - (0,1)\cdot\left(\frac{2}{7}, \frac{18}{7}\right)^T}{(0,1)\cdot\left(\frac{3}{14}, -\frac{9}{28}\right)^T} = \frac{-\frac{18}{7}}{-\frac{9}{28}} = 8. \end{aligned}$$
故
$$\alpha_2 = \min\{1, \bar{\alpha}\} = 1.$$
令
$$x_3 = x_2 + \alpha_2 d_2 = \left(\frac{2}{7}, \frac{18}{7}\right)^T + 1\cdot\left(\frac{3}{14}, -\frac{9}{28}\right)^T = \left(\frac{1}{2}, \frac{9}{4}\right)^T.$$
计算出 $g_3 = \nabla f(x_3) = \left(-\frac{9}{4}, -\frac{3}{2}\right)^T$.

在点 x_3 处, 计算相应的拉格朗日乘子. 此时 $S_3 = \{1\}$, $A_3 = (-3, -2)$, 那么,
$$\lambda_1^3 = [(A_3 H^{-1} A_3^T)^{-1} A_3 H^{-1}] g_3 = \frac{3}{4} > 0.$$
因此 $x_3 = \left(\frac{1}{2}, \frac{9}{4}\right)^T$ 是所求的最优解.

11.2.3　有效集方法的 Matlab 程序

由于有效集方法是求解凸二次规划问题的一种值得推荐的方法, 本节给出有效集方法的 Matlab 程序. 在实际使用有效集方法求解凸二次规划时, 一般用渐进有效集约束指标集代替有效集约束指标集, 即取 $\{i \in I \,|\, a_i^T x - b_i \le \varepsilon\}$ 近似代替 $I(x^*)$, 其中 $\varepsilon > 0$ 是比较小的常数. 这样做的好处是使得迭代步长不至于太短. 此外, 算法 31 还需要确立一个初始可行点. 可采用下述方法: 给出一个初始估计点 $\bar{x} \in \mathbb{R}^n$, 定义下列线性规划

$$\begin{cases} \min\ e^T z, \\ \text{s.t.}\ \ a_i^T x + \tau_i z_i - b_i = 0,\ i \in E = \{1, \cdots, l\}, \\ \qquad a_i^T x + z_i - b_i \ge 0,\ i \in I = \{l+1, \cdots, m\}, \\ \qquad z_1 \ge 0, \cdots, z_m \ge 0, \end{cases} \tag{11.21}$$

其中, $e = (1, \cdots, 1)^T$, $\tau_i = -\text{sign}(a_i^T \bar{x} - b_i)$, $i \in E$. 问题 (11.21) 的一个初始可行点为

$$x = \bar{x},\ \ z_i = |a_i^T \bar{x} - b_i|\,(i \in E);\ z_i = \max\{b_i - a_i^T \bar{x},\, 0\}\,(i \in I).$$

不难证明, 如果 $\tilde{x}$ 是问题 (11.15) 的可行点, 那么 $(\tilde{x}, 0)$ 是子问题 (11.21) 的最优解. 反之, 如果问题 (11.15) 有可行点, 则问题 (11.21)的最优值为 0, 从而子问题 (11.21) 的任何一个解产生问题 (11.15) 的一个可行点.

下面给出用有效集方法求解一般凸二次规划问题的 Matlab 程序, 在某种意义下, 该程序是通用的.

程序 17　本程序主要适用于求解一般约束条件下的凸二次规划问题.

```
function [x,lamk,exitflag,output]=qpact(H,c,Ae,be,Ai,bi,x0)
%功能: 用有效集方法解一般约束二次规划问题:
%    min f(x)=0.5*x'*H*x+c'*x,
%    s.t.  a'_i*x-b_i=0,(i=1,...,l),
%          a'_i*x-b_i>=0,(i=l+1,...,m)
%输入: x0是初始点, H, c分别是目标函数二次型矩阵和向量;
%    Ae=(a_1,...,a_l)',  be=(b_1,...,b_l)';
%    Ai=(a_{l+1},...,a_m), bi=(b_{l+1},...,b_m)'.
%输出: x是最优解,  lambda是对应的乘子向量; output是结构变量,
%       输出极小值f(x), 迭代次数k等信息, exitflag是算法终止类型
```

```
%%%%%%%%%%%%%%%%% 主程序开始  %%%%%%%%%%%%%%%%%
% 初始化
epsilon=1.0e-9; err=1.0e-6;
k=0; x=x0;  n=length(x);  kmax=1.0e3;
ne=length(be); ni=length(bi); lamk=zeros(ne+ni,1);
index=ones(ni,1);
for (i=1:ni)
    if(Ai(i,:)*x>bi(i)+epsilon), index(i)=0; end
end
%算法主程序
while (k<=kmax)
    %求解子问题
    Aee=[];
    if(ne>0), Aee=Ae; end
    for(j=1:ni)
        if(index(j)>0), Aee=[Aee; Ai(j,:)]; end
    end
    gk=H*x+c;
    [m1,n1] = size(Aee);
    [dk,lamk]=qsubp(H,gk,Aee,zeros(m1,1));
    if(norm(dk)<=err)
        y=0.0;
        if(length(lamk)>ne)
            [y,jk]=min(lamk(ne+1:length(lamk)));
        end
        if(y>=0)
            exitflag=0;
        else
            exitflag=1;
            for(i=1:ni)
                if(index(i)&(ne+sum(index(1:i)))==jk)
                    index(i)=0; break;
                end
            end
        end
        k=k+1;
    else
```

```
            exitflag=1;
            %求步长
            alpha=1.0; tm=1.0;
            for(i=1:ni)
                if((index(i)==0)&(Ai(i,:)*dk<0))
                    tm1=(bi(i)-Ai(i,:)*x)/(Ai(i,:)*dk);
                    if(tm1<tm)
                        tm=tm1; ti=i;
                    end
                end
            end
            alpha=min(alpha,tm);
            x=x+alpha*dk;
            %修正有效集
            if(tm<1), index(ti)=1; end
        end
        if(exitflag==0), break; end
        k=k+1;
    end
    output.fval=0.5*x'*H*x+c'*x;
    output.iter=k;
    %%%%%%%% 求解子问题 %%%%%%%%%%%%%%
    function [x,lambda]=qsubp(H,c,Ae,be)
    ginvH=pinv(H);
    [m,n]=size(Ae);
    if(m>0)
        rb=Ae*ginvH*c + be;
        lambda=pinv(Ae*ginvH*Ae')*rb;
        x=ginvH*(Ae'*lambda-c);
    else
        x=-ginvH*c;
        lambda=0;
    end
```

注 (1) 关于上述程序, 在子函数 qsubp 中, 使用的是广义逆 (pinv 是 Matlab 软件内置的求广义逆的函数), 这样不仅可计算 H 是奇异阵的情形, 同时保证了计算的数值稳定性. 另外, 该子函数还包含了子问题为无约束二次规划时的解. (2) 使用程序 17 时, 需要用户提供所有问题的目标函数和约束函数的有关数据,

可通过编制一个 m 文件来解决.

例 34　利用程序 17 重新求解例 33, 即

$$\begin{cases} \min\ f(x)=x_1^2-x_1x_2+2x_2^2-x_1-10x_2, \\ \text{s.t.}\ -3x_1-2x_2\ge -6, \\ \qquad x_1\ge 0,\ x_2\ge 0. \end{cases}$$

解　首先确定有关数据:

$$H=\begin{bmatrix} 2 & -1 \\ -1 & 4 \end{bmatrix},\ c=\begin{bmatrix} -1 \\ -10 \end{bmatrix},\ \text{Ae}=[\,],\ \text{be}=[\,], \text{Ai}=\begin{bmatrix} -3 & -2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},\ \text{bi}=\begin{bmatrix} -6 \\ 0 \\ 0 \end{bmatrix}.$$

编制一个利用上述数据调用程序 17 的函数文件 callqpact.m

```
function callqpact
H=[2 -1; -1 4];
c=[-1 -10]';
Ae=[ ];  be=[ ];
Ai=[-3 -2; 1 0; 0 1];
bi=[-6 0 0]';
x0=[0 0]';
[x,lambda,exitflag,output]=qpact(H,c,Ae,be,Ai,bi,x0)
```

然后在 Matlab 命令窗口键入 callqpact, 回车即得结果

```
x =
    0.5000
    2.2500
lambda =
    0.7500
exitflag =
    0
output =
    fval: -13.7500
    iter: 8
```

可以看出, 上述结果跟例 33 用手算是一致的.

例 35　利用程序 17 求解下列二次规划问题

$$\begin{cases} \min\ f(x) = \dfrac{1}{2}x_1^2 - x_1x_2 + x_2^2 - 6x_1 - 2x_2, \\ \text{s.t.}\ -2x_1 - x_2 \geq -3, \\ \qquad x_1 - x_2 \geq -1, \\ \qquad -x_1 - 2x_2 \geq -2, \\ \qquad x_1 \geq 0,\ x_2 \geq 0. \end{cases}$$

该问题有精确解 $x^* = \left(\frac{4}{3}, \frac{1}{3}\right)^T$, 最优值 $f(x^*) = -8\frac{1}{9}$.

解　在 Matlab 命令窗口输入下列命令:

```
H=[1 -1;-1 2];
c=[-6 -2]';
Ai=[-2 -1;1 -1;-1 -2; 1 0;0 1];
bi=[-3 -1 -2 0 0]';
x=[0 0]';
[x,lambda,exitflag,output]=qpact(H,c,[],[],Ai,bi,x0)
```

得计算结果为

```
x =
    1.3333
    0.3333
lambda =
    2.4444
    0.1111
exitflag =
    0
output =
    fval: -8.1111
    iter: 7
```

习 题 11

1. 用拉格朗日方法求解下列二次规划问题:

(1) $\min\ f(x)=2x_1^2+x_2^2+x_1x_2-x_1-x_2,$

$\text{s.t.}\ x_1+x_2=1.$

(2) $\min\ f(x)=\frac{3}{2}x_1^2-x_1x_2+x_2^2-x_2x_3+\frac{1}{2}x_3^2+x_1+x_2+x_3,$

$\text{s.t.}\ x_1+2x_2+x_3=4.$

2. 用有效集方法求解下列二次规划问题:

(1) $\min\ f(x)=9x_1^2+9x_2^2-30x_1-72x_2,$

$\text{s.t.}\ -2x_1-x_2\geq -4,$

$x_1,\ x_2\geq 0.$

(2) $\min\ f(x)=x_1^2-x_1x_2+x_2^2-3x_1,$

$\text{s.t.}\ -x_1-x_2\geq -2,$

$x_1,\ x_2\geq 2.$

3. 证明: 若矩阵

$$M=\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix}$$

非奇异, 则 A 必是行满秩的.

4. 设矩阵 $W\in\mathbb{R}^{n\times n}$ 对称, $Z\in\mathbb{R}^{n\times l}$, $u\in\mathbb{R}^n$, 记 $\bar{Z}=(Z,u)$, 试证明：若矩阵 Z^TWZ 正定, 则矩阵 $\bar{Z}^TW\bar{Z}$ 半正定.

5. 设 $A\in\mathbb{R}^{m\times n}$ 行满秩, $a\in\mathbb{R}^n$, 证明: 二次规划问题

$$\begin{aligned} \min\ & \frac{1}{2}(x-a)^T(x-a), \\ \text{s.t.}\ & Ax=b \end{aligned}$$

的解以及相应的拉格朗日乘子分别为:

$$x^*=a+A^T(AA^T)^{-1}(b-Aa),\ \ \lambda^*=(AA^T)^{-1}(b-Aa).$$

6. 设 H 对称正定, λ^* 是问题

$$\begin{aligned} \min\ & \frac{1}{2}\lambda^T(A^TH^{-1}A)\lambda-(b+A^TH^{-1}c)^T\lambda, \\ \text{s.t.}\ & \lambda_i\geq 0,\ i\in I \end{aligned}$$

的解. 证明: $x^*=-H^{-1}(c-A\lambda^*)$ 是问题

$$\begin{aligned} \min\ & f(x)=\frac{1}{2}x^THx+c^Tx, \\ \text{s.t.}\ & a_i^Tx\geq b_i,\ i\in I=\{1,\cdots,m_1\}, \\ & a_i^Tx=b_i,\ i\in E=\{m_1+1,\cdots,m\} \end{aligned}$$

的最优解.

7. 设 $d_k \neq 0$ 是问题

$$\begin{aligned} \min \quad & \frac{1}{2}d^T H d + \nabla f(x_k)^T d, \\ \text{s.t.} \quad & a_i^T d = 0,\ i \in S(x_k) \end{aligned}$$

的最优解, 且二阶充分条件成立, 其中 $S(x_k) = \{i \in I \cup E : a_i^T x_k = b_i\}$, 证明:

$$f(x_k + \alpha d_k) < f(x_k),\ \forall \alpha \in (0, 1].$$

8. 设 $H \in \mathbb{R}^{n\times n}$ 对称正定, $A \in \mathbb{R}^{m\times n}$ 行满秩.

(1) 证明矩阵

$$M = \begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix}$$

非奇异, 且

$$M^{-1} = \begin{pmatrix} C & E \\ E^T & F \end{pmatrix},$$

其中

$$\begin{aligned} & C = H^{-1} - H^{-1}A^T(AH^{-1}A^T)^{-1}AH^{-1}, \\ & E = H^{-1}A^T(AH^{-1}A^T)^{-1},\ \ F = -(AH^{-1}A^T)^{-1}. \end{aligned}$$

(2) 证明: 矩阵 M 有 n 个正特征值, m 个负特征值, 没有 0 特征值.

(3) 证明线性方程组

$$M\begin{pmatrix} -p \\ \lambda \end{pmatrix} = \begin{pmatrix} g \\ b \end{pmatrix}$$

等价于下面的线性方程组

$$(AH^{-1}A^T)\lambda = AH^{-1}g - b,\ Hp = A^T\lambda - g.$$

第十二章　序列二次规划法

本章考虑求解一般非线性优化问题

$$\begin{cases} \min\ f(x), \\ \text{s.t.}\ \ h_i(x) = 0,\ i \in E = \{1, \cdots, l\}, \\ \qquad g_i(x) \geq 0,\ i \in I = \{1, \cdots, m\} \end{cases} \tag{12.1}$$

的序列二次规划法 (SQP, Sequential Quadratic Programming). SQP 方法是求解约束优化问题最有效的算法之一, 其基本思想是: 在每一迭代步通过求解一个二次规划子问题来确立一个下降方向, 以减少价值函数来取得步长, 重复这些步骤直到求得原问题的解. 这也是之所以称为序列二次规划法的由来. 本章主要介绍 SQP 方法的基本思想、迭代步骤和收敛性分析. 首先介绍求解等式约束优化问题的牛顿-拉格朗日法.

12.1　牛顿-拉格朗日法

12.1.1　牛顿-拉格朗日法的基本理论

考虑纯等式约束的优化问题

$$\begin{cases} \min\ f(x), \\ \text{s.t.}\ \ h_i(x) = 0,\ i \in E = \{1, \cdots, l\}, \end{cases} \tag{12.2}$$

其中 $f : \mathbb{R}^n \to \mathbb{R}$, $h_i : \mathbb{R}^n \to \mathbb{R}\,(i \in E)$ 都是二阶连续可微的实函数. 记 $h(x) = (h_1(x), \cdots, h_l(x))^T$, 则不难写出该问题的拉格朗日函数为

$$L(x, \mu) = f(x) - \sum_{i=1}^{l} \mu_i h_i(x) = f(x) - \mu^T h(x),$$

其中 $\mu=(\mu_1,\cdots,\mu_l)^T$ 为拉格朗日乘子向量. 约束函数 $h(x)$ 的梯度矩阵为

$$\nabla h(x)=\big[\nabla h_1(x),\cdots,\nabla h_l(x)\big],$$

则 $h(x)$ 的 Jacobi 矩阵为 $A(x)=\nabla h(x)^T$. 根据问题 (12.2) 的 KT 条件, 可以得到如下的方程组

$$\nabla L(x,\mu)=\begin{bmatrix}\nabla_x L(x,\mu)\\ \nabla_\mu L(x,\mu)\end{bmatrix}=\begin{bmatrix}\nabla f(x)-A(x)^T\mu\\ -h(x)\end{bmatrix}=0. \tag{12.3}$$

现在考虑用牛顿法求解上述的非线性方程组 (12.3). 记函数 $\nabla L(x,\mu)$ 的 Jacobi 矩阵为

$$N(x,\mu)=\begin{bmatrix}W(x,\mu) & -A(x)^T\\ -A(x) & 0\end{bmatrix}, \tag{12.4}$$

其中

$$W(x,\mu)=\nabla^2_{xx}L(x,\mu)=\nabla^2 f(x)-\sum_{i=1}^{l}\mu_i\nabla^2 h_i(x)$$

是拉格朗日函数 $L(x,\mu)$ 关于 x 的 Hesse 阵. (12.4) 所定义的矩阵 $N(x,\mu)$ 亦称之为 KT 矩阵. 对于给定的点 $z_k=(x_k,\mu_k)$, 牛顿法的迭代格式为

$$z_{k+1}=z_k+p_k, \tag{12.5}$$

其中 $p_k=(d_k,\nu_k)$ 满足下面的线性方程组:

$$N(x_k,\mu_k)p_k=-\nabla L(x_k,\mu_k),$$

即

$$\begin{bmatrix}W(x_k,\mu_k) & -A(x_k)^T\\ -A(x_k) & 0\end{bmatrix}\begin{bmatrix}d_k\\ \nu_k\end{bmatrix}=\begin{bmatrix}-\nabla f(x_k)+A(x_k)^T\mu_k\\ h(x_k)\end{bmatrix}. \tag{12.6}$$

不难发现, 只要矩阵 $A(x_k)$ 行满秩且 $W(x_k,\mu_k)$ 是正定的, 那么方程组 (12.6) 的系数矩阵是非奇异的, 且该方程有唯一解. 由于 KT 条件 (12.3) 是拉格朗日函数稳定点的条件, 所以人们通常把基于求解方程 (12.3) 的优化方法称为拉格朗日方法. 特别地, 如果用牛顿法求解该方程组, 则称之为牛顿-拉格朗日方法. 因此, 根据牛顿法的性质, 该方法具有局部二次收敛性质.

下面写出牛顿-拉格朗日方法的详细算法步骤.

算法 32 (牛顿-拉格朗日方法)

步 0　选取 $x_0 \in \mathbb{R}^n$, $\mu_0 \in \mathbb{R}^l$, $\rho, \gamma \in (0,1)$, $0 \le \varepsilon \ll 1$. 令 $k := 0$.

步 1　计算 $\|\nabla L(x_k, \mu_k)\|$ 的值. 若 $\|\nabla L(x_k, \mu_k)\| \le \varepsilon$, 停算. 否则, 转步 2.

步 2　解方程组 (12.6) 得 $p_k = (d_k, \nu_k)$.

步 3　若

$$\|\nabla L(x_k + d_k, \mu_k + \nu_k)\|^2 \le (1-\gamma)\|\nabla L(x_k, \mu_k)\|^2, \tag{12.7}$$

则置 $\alpha_k := 1$, 转步 5; 否则, 转步 4.

步 4　令 m_k 是使下面的不等式成立的最小非负整数 m :

$$\|\nabla L(x_k + \rho^m d_k, \mu_k + \rho^m \nu_k)\|^2 \le (1-\gamma\rho^m)\|\nabla L(x_k, \mu_k)\|^2, \tag{12.8}$$

置 $\alpha_k = \rho^{m_k}$.

步 5　令 $x_{k+1} = x_k + \alpha_k d_k$, $\mu_{k+1} = \mu_k + \alpha_k \nu_k$. 置 $k := k+1$, 转步 1.

12.1.2　牛顿-拉格朗日法的 Matlab 程序

本小节通过一个具体的例子来介绍算法 32 (牛顿-拉格朗日方法) 的 Matlab 实现.

例 36　用算法 32 编程计算下列最优化问题的极小点

$$\begin{cases} \min & f(x) = e^{x_1x_2x_3x_4x_5} - \dfrac{1}{2}(x_1^3 + x_2^3 + 1)^2, \\ \text{s.t.} & x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0, \\ & x_2x_3 - 5x_4x_5 = 0, \\ & x_1^3 + x_2^3 + 1 = 0. \end{cases}$$

该问题有最优解 $x^* = (-1.71, 1.59, 1.82, -0.763, -0.763)^T$, 最优值 $f(x^*) = 0.0539$.

解　编制 Matlab 程序如下

```
function [x,mu,val,mh,k]=newtlagr(x0,mu0)
% 功能:用牛顿-拉格朗日法求解约束优化问题:
%     min f(x), s.t. h_i(x)=0, i=1,..., l.
%输入:x0是初始点, mu0是乘子向量的初始值
```

```
%输出: x,mu分别是近似最优点及相应的乘子,
%val是最优值,mh是约束函数的模,k是迭代次数.
maxk=500; %最大迭代次数
n=length(x0); l=length(mu0);
rho=0.5; gamma=0.4;
x=x0; mu=mu0;
k=0;  epsilon=1e-8;
while(k<maxk)
    dl=dla(x,mu);  %计算乘子函数的梯度
    if(norm(dl)<epsilon), break; end  %检验终止准则
    N=N1(x,mu);   % 计算拉格朗日矩阵
    dz=-N\dl;    %解方程组得搜索方向
    dx=dz(1:n); du=dz(n+1:n+l);
    m=0; mk=0;
    while(m<20)   % Armijo搜索
        if(norm(dla(x+rho^m*dx,mu+rho^m*du))^2<=(1-gamma*rho^m)*norm(dl)^2)
            mk=m; break;
        end
        m=m+1;
    end
    x=x+rho^mk*dx; mu=mu+rho^mk*du;
    k=k+1;
end
val=f1(x);
mh=norm(h1(x),inf);
%%%%%%%%% 拉格朗日函数 L(x,mu) %%%%%%%%%%%%%
function l=la(x,mu)
f=f1(x);   %调用目标函数文件
h=h1(x);  %调用约束函数文件
l=f-mu'*h; % 计算乘子函数
%%%%%%%%% 拉格朗日函数的梯度 %%%%%%%%%%%%
function dl=dla(x,mu)
df=df1(x); %调用目标函数梯度文件
h=h1(x);  %调用约束函数文件
dh=dh1(x);  %调用约束函数Jacobi矩阵文件
dl=[df-dh'*mu; -h];  %计算乘子函数梯度文件
%%%%%%%%% 拉格朗日函数的Hesse阵 %%%%%%%%%%
```

```
function d2l=d2la(x,mu)
d2f=d2f1(x);   %调用目标函数Hesse阵文件
[d2h1,d2h2,d2h3]=d2h(x); %调用约束函数二阶导数文件
d2l=d2f-mu(1)*d2h1-mu(2)*d2h2-mu(3)*d2h3; %计算乘子函数的Hesse阵
%%%%%%%%% 系数矩阵N(x,mu) %%%%%%%%%%%
function N=N1(x,mu)  %计算拉格朗日矩阵
l=length(mu);
d2l=d2la(x,mu);   dh=dh1(x);
N=[d2l, -dh'; -dh, zeros(l,l)];
%%%%%%%%% 目标函数 f(x) %%%%%%%%%%%%
function f=f1(x)
s=x(1)*x(2)*x(3)*x(4)*x(5);
f=exp(s)-0.5*(x(1)^3+x(2)^3+1)^2;
%%%%%%%%% 约束函数 h(x) %%%%%%%%%%%%
function h=h1(x)
h=[x(1)^2+x(2)^2+x(3)^2+x(4)^2+x(5)^2-10;x(2)*x(3)-5*x(4)*x(5);x(1)^3+x(2)^3+1];
%%%%%%%%% 目标函数 f(x) 的梯度%%%%%%%%%%%%
function df=df1(x)
s=x(1)*x(2)*x(3)*x(4)*x(5);
df(1)=s/(x(1))*exp(s)-3*(x(1)^3+x(2)^3+1)*x(1)^2;
df(2)=s/(x(2))*exp(s)-3*(x(1)^3+x(2)^3+1)*x(2)^2;
df(3)=s/(x(3))*exp(s); df1(4)=s/(x(4))*exp(s);
df(5)=s/(x(5))*exp(s);
df=df(:);
%%%%%%%%% 约束函数 h(x) 的Jacobi矩阵A(x)%%%%%%%%%%%%
function dh=dh1(x)
dh=[2*x(1),2*x(2),2*x(3),2*x(4),2*x(5);0,x(3),x(2),-5*x(5),-5*x(4);...
    3*x(1)^2,3*x(2)^2,0,0,0];
%%%%%%%%% 目标函数 f(x) 的Hesse阵%%%%%%%%%%%%
function d2f=d2f1(x)
s=x(1)*x(2)*x(3)*x(4)*x(5);
d2f=[(s/(x(1)))^2*exp(s)-6*x(1)*(x(1)^3+x(2)^3+1)-9*x(1)^4,...
    (1+s)*x(3)*x(4)*x(5)*exp(s)-9*x(1)^2*x(2)^2,(1+s)*x(2)*x(4)*x(5)*exp(s),...
    (1+s)*x(2)*x(3)*x(5)*exp(s), (1+s)*x(2)*x(3)*x(4)*exp(s); ...
    (1+s)*x(3)*x(4)*x(5)*exp(s)-9*x(1)^2*x(2)^2, ...
    (s/(x(2)))^2*exp(s)-6*x(2)*(x(1)^3+x(2)^3+1)-9*x(2)^4,...
    (1+s)*x(1)*x(4)*x(5)*exp(s),(1+s)*x(1)*x(3)*x(5)*exp(s),...
```

```
    (1+s)*x(1)*x(3)*x(4)*exp(s);...
    (1+s)*x(2)*x(4)*x(5)*exp(s),(1+s)*x(1)*x(4)*x(5)*exp(s),s^2/(x(3))*exp(s),...
    (1+s)*x(1)*x(2)*x(5)*exp(s),(1+s)*x(1)*x(2)*x(4)*exp(s); ...
    (1+s)*x(2)*x(3)*x(5)*exp(s),(1+s)*x(1)*x(3)*x(5)*exp(s), ...
    (1+s)*x(1)*x(2)*x(5)*exp(s),s^2/(x(4))*exp(s),(1+s)*x(1)*x(2)*x(3)*exp(s);...
    (1+s)*x(2)*x(3)*x(4)*exp(s),(1+s)*x(1)*x(3)*x(4)*exp(s), ...
    (1+s)*x(1)*x(2)*x(4)*exp(s),(1+s)*x(1)*x(2)*x(3)*exp(s),s^2/(x(5))*exp(s)]';
%%%%%%%%% 约束函数 h(x) 的Hesse阵%%%%%%%%%%%%
function [d2h1,d2h2,d2h3]=d2h(x)
d2h1=[2 0 0 0 0; 0 2 0 0 0; 0 0 2 0 0; 0 0 0 2 0; 0 0 0 0 2]';
d2h2=[0 0 0 0 0; 0 0 1 0 0; 0 1 0 0 0; 0 0 0 0 -5; 0 0 0 -5 0]';
d2h3=[6*x(1) 0 0 0 0;0 6*x(2) 0 0 0;0 0 0 0 0;0 0 0 0 0;0 0 0 0 0]';
```

利用上面的程序, 取乘子向量的初值为 $\mu_0 = (0,0,0)$, 终止准则值取为 $\|\nabla L(x_k,\mu_k)\|^2 \le 10^{-12}$, 对于不同的初始点得到计算结果如下表所示.

初始点 (x_0)	迭代次数 (k)	$f(x_k)$ 的值	$\|h(x_k)\|$ 的值
$(-1.71.51.8-0.6-0.6)^T$	11	0.0539	3.2894e-011
$(-1.7,1.6,1.8,-0.7,-0.7)^T$	11	0.0539	5.9272e-012
$(-1.8,1.7,1.9,-0.8,-0.8)^T$	9	0.0539	3.0606e-011
$(-2,1.5,2,-1,-1)^T$	14	0.0539	1.4766e-011
$(-3,2,3,-2,-2)^T$	16	0.0539	3.6545e-011

说明 Matlab 调用方式为: 在命令窗口依次输入如下命令并回车即得计算结果.

```
x0=[-1.7,1.6,1.8,-0.7,-0.7]';
mu0=[0.1 0.1 0.1]';
[x,mu,val,mh,k]=newtlagr(x0,mu0)
```

12.2　SQP 方法的算法模型

12.2.1　基于拉格朗日函数 Hesse 阵的 SQP 方法

前一节介绍的牛顿-拉格朗日法, 由于每一迭代步求解方程组 (12.6) 数值上不是很稳定, 因此这一方法并不实用. 但它有一个重要的贡献, 就是以它为基础

发展了序列二次规划方法 (SQP 方法). 鉴于方程组 (12.6) 的求解数值不稳定, 故考虑将它转化为一个严格凸二次规划问题. 转化的条件是 (12.2) 的解点 x^* 处最优性二阶充分条件成立, 即对满足 $A(x^*)^T d = 0$ 的任一向量 $d \neq 0$, 成立

$$d^T W(x^*, \mu^*) d > 0.$$

这时, 由引理 16 知, 当 $\tau > 0$ 充分小时, 有

$$W(x^*, \mu^*) + \frac{1}{2\tau} A(x^*)^T A(x^*)$$

正定. 考虑 (12.6) 中的 $W(x_k, \mu_k)$ 用一个正定矩阵来代替, 记

$$B(x_k, \mu_k) = W(x_k, \mu_k) + \frac{1}{2\tau} A(x_k)^T A(x_k),$$

则当 $(x_k, \mu_k) \to (x^*, \mu^*)$ 时, 矩阵 $B(x^*, \mu^*)$ 正定. 注意到 (12.6) 的第一个展开式为

$$W(x_k, \mu_k) d_k - A(x_k)^T \nu_k = -\nabla f(x_k) + A(x_k)^T \mu_k,$$

将上式变形为

$$\left[W(x_k, \mu_k) + \frac{1}{2\tau} A(x_k)^T A(x_k)\right] d_k - A(x_k)^T \left[\mu_k + \nu_k + \frac{1}{2\tau} A(x_k) d_k\right] = -\nabla f(x_k).$$

令

$$\bar{\mu}_k := \mu_k + \nu_k + \frac{1}{2\tau} A(x_k) d_k,$$

即得

$$B(x_k, \mu_k) d_k - A(x_k)^T \bar{\mu}_k = -\nabla f(x_k).$$

因此 (12.6) 等价于

$$\begin{bmatrix} B(x_k, \mu_k) & -A(x_k)^T \\ A(x_k) & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \bar{\mu}_k \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) \\ h(x_k) \end{bmatrix}. \tag{12.9}$$

进一步, 可以把方程 (12.9) 转化为严格凸二次规划. 我们有下面的定理:

定理 59 设 $B(x_k, \mu_k)$ 是 $n \times n$ 正定矩阵, $A(x_k)$ 是 $m \times n$ 行满秩矩阵. 则 d_k 满足 (12.9) 的充要条件是 d_k 是下列严格凸二次规划

$$\begin{cases} \min \quad q_k(d) = \dfrac{1}{2} d^T B(x_k, \mu_k) d + \nabla f(x_k)^T d \\ \text{s.t.} \quad h(x_k) + A(x_k) d = 0 \end{cases} \tag{12.10}$$

的全局极小点.

证 设 d_k 是 (12.10) 的全局极小点, 注意到 $A(x_k)$ 行满秩, 故由 KT 条件, 存在乘子向量 $\bar{\mu}_k$, 使得

$$\nabla f(x_k) + B(x_k, \mu_k)d_k - A(x_k)^T\bar{\mu}_k = 0.$$

再由 (12.10) 的约束条件知, $(d_k, \bar{\mu}_k)$ 是方程组 (12.9) 的解.

反之, 设 $(d_k, \bar{\mu}_k)$ 是方程组 (12.9) 的解. 由于 $B(x_k, \mu_k)$ 正定, $A(x_k)$ 行满秩, 故方程组 (12.9) 的系数矩阵是非奇异的, 从而这个解是唯一的. 由定理 56 知, $(d_k, \bar{\mu}_k)$ 是 (12.10) 的 KT 对, 从而 d_k 是 (12.10) 的全局极小点. □

为了方便, 定义罚函数

$$P(x, \mu) = \|\nabla L(x, \mu)\|^2 = \|\nabla f(x) - A(x)^T\mu\|^2 + \|h(x)\|^2. \tag{12.11}$$

不难证明, 由 (12.6) 确定的 p_k 满足 (参见文献 [12])

$$\nabla P(x_k, \mu_k)^T p_k = -2P(x_k, \mu_k) \le 0. \tag{12.12}$$

我们有下面的算法:

算法 33 (**纯等式约束优化问题的 SQP 方法**)

步 0 选取 $x_0 \in \mathbb{R}^n$, $\mu_0 \in \mathbb{R}^l$, ρ, $\gamma \in (0,1)$, $0 \le \varepsilon \ll 1$. 令 $k := 0$.

步 1 计算 $P(x_k, \mu_k)$ 的值. 若 $P(x_k, \mu_k) \le \varepsilon$, 停算. 否则, 转步 2.

步 2 求解二次规划子问题 (12.10) 得 d_k 和 $\bar{\mu}_k$, 并置

$$\nu_k = \bar{\mu}_k - \mu_k - \frac{1}{2\tau}A(x_k)d_k.$$

步 3 若

$$P(x_k + d_k, \mu_k + \nu_k) \le (1-\gamma)P(x_k, \mu_k), \tag{12.13}$$

则置 $\alpha_k := 1$, 转步 5; 否则, 转步 4.

步 4 令 m_k 是使下面的不等式成立的最小非负整数 m:

$$P(x_k + \rho^m d_k, \mu_k + \rho^m \nu_k) \le (1-\gamma\rho^m)P(x_k, \mu_k), \tag{12.14}$$

置 $\alpha_k = \rho^{m_k}$.

步 5 令 $x_{k+1} = x_k + \alpha_k d_k$, $\mu_{k+1} = \mu_k + \alpha_k \nu_k$. 置 $k := k+1$, 转步 1.

不难发现, 在上面的算法中, 若 $\alpha_k < 1$, 则必有

$$P(x_k + \rho^{m_k-1}d_k, \mu_k + \rho^{m_k-1}\nu_k) > (1 - \gamma\rho^{m_k-1})P(x_k, \mu_k). \tag{12.15}$$

下面给出算法 33 的全局收敛性定理.

定理 60 对于等式约束问题 (12.2), 若 SQP 算法 33 生成的迭代序列 $\{(x_k, \mu_k)\}$ 使得 KT 矩阵的逆矩阵 $N(x_k, \mu_k)^{-1}$ 一致有界, 则 $\{(x_k, \mu_k)\}$ 的任何聚点 (x^*, μ^*) 都满足 $P(x^*, \mu^*) = 0$. 特别地, $\{x_k\}$ 的任一聚点都是问题 (12.2) 的 KT 点.

证 用反证法. 不失一般性, 假定 $\{(x_k, \mu_k)\} \to (x^*, \mu^*)$. 若 $P(x^*, \mu^*) > 0$. 由步 3 和步 4 可知

$$P(x_{k+1}, \mu_{k+1}) \le (1 - \gamma\alpha_k)P(x_k, \mu_k) < P(x_k, \mu_k).$$

由上式及 $P(x_k, \mu_k) \to P(x^*, \mu^*) > 0$ 可推得

$$\lim_{k\to\infty} \alpha_k = 0.$$

另外, 由 (12.6) 可得

$$p_k = \begin{bmatrix} d_k \\ \nu_k \end{bmatrix} = -N(x_k, \mu_k)^{-1}\nabla L(x_k, \mu_k).$$

注意到矩阵 $N(x_k, \mu_k)^{-1}$ 的一致有界性, $p_k = (d_k, \nu_k)$ 也是一致有界的, 且 $p_k \to p^* = (d^*, \nu^*)$, 其中 p^* 满足牛顿方程 $N(x^*, \mu^*)p^* = -\nabla L(x^*, \mu^*)$. 由于 $\alpha_k' = \alpha_k/\rho = \rho^{m_k-1} \to 0$. 故由 (12.15) 有

$$\frac{P(x_k + \alpha_k' d_k, \mu_k + \alpha_k' \nu_k) - P(x_k, \mu_k)}{\alpha_k'} > -\gamma P(x_k, \mu_k).$$

对上式两边取极限得

$$-\gamma P(x^*, \mu^*) \le \nabla P(x^*, \mu^*)^T p^* = -2P(x^*, \mu^*),$$

即

$$(2 - \gamma)P(x^*, \mu^*) \le 0.$$

注意到 $0<\gamma<1$, 故得 $P(x^*,\mu^*)\le 0$, 这与假设 $P(x^*,\mu^*)>0$ 矛盾. 因此必有 $P(x^*,\mu^*)=0$.

现在来证明定理的第二部分. 设 x^* 是序列 $\{x_k\}$ 的任一聚点, 不失一般性, 假设 $\{x_k\}\to x^*$. 注意到序列 $\{P(x_k,\mu_k)\}$ 是单调不增的且矩阵序列 $\{N(x_k,\mu_k)^{-1}\}$ 是一致有界的, 由此可推得 $\alpha_k\to 0\,(k\to\infty)$, 且 $\{p_k=(d_k,\nu_k)\}$ 是有界的, 故 $\{d_k\}$ 和 $\{\nu_k\}$ 都是有界的. 又由步 2, 严格凸二次规划子问题的解 $\{(d_k,\bar{\mu}_k)\}$ 也是有界的, 故 $\{\bar{\mu}_k\}$ 是有界的, 即 $\{\bar{\mu}_k=\mu_k+\nu_k\}$ 是有界的. 特别地, 对于充分大的 k, $\{\mu_{k+1}:=\mu_k+\alpha_k\nu_k\}$ 是有界的, 于是存在收敛的子序列. 假设 μ^* 是 $\{\mu_{k+1}\}$ 的某个聚点, 则 (x^*,μ^*) 是迭代序列 $\{(x_k,\mu_k)\}$ 的一个聚点. 由第一部分的证明可知 $P(x^*,\mu^*)=0$, 因此, x^* 是问题 (12.2) 的 KT 点. □

关于算法 33 的收敛速度, 我们有下面的定理, 其证明过程可参见文献 [1], 兹略去不证.

定理 61　设算法 33 产生的迭代序列 $\{x_k\}$ 收敛到一个局部极小点 x^*. 若函数 f, $h=(h_1,\cdots,h_l)^T$ 在 x^* 附近三阶连续可微, Jacobi 矩阵 $A(x^*)=\nabla h(x)^T$ 行满秩, 且二阶最优性充分条件成立. 则有

(1)　必有 $\{\mu_k\}\to\mu^*$, 其中 μ^* 是等式约束优化问题 (12.6) 的拉格朗日乘子, 且整个迭代序列 $\{(x_k,\mu_k)\}$ 是二阶收敛的, 即

$$\|(x_{k+1}-x^*,\ \mu_{k+1}-\mu^*)\|=O(\|(x_k-x^*,\ \mu_k-\mu^*)\|^2).$$

(2)　序列 $\{x_k\}$ 超线性收敛到 x^*, 且

$$\|x_{k+1}-x^*\|=o\Big(\|x_k-x^*\|\prod_{i=1}^{t}\|x_{k-i}-x^*\|\Big),$$

其中 t 是任意给定的正整数.

例 37　编制 Matlab 程序, 用算法 33 计算下列最优化问题的极小点

$$\begin{cases}\min & f(x)=\mathrm{e}^{x_1x_2x_3x_4x_5}-\dfrac{1}{2}(x_1^3+x_2^3+1)^2,\\[2mm] \text{s.t.} & x_1^2+x_2^2+x_3^2+x_4^2+x_5^2-10=0,\\[2mm] & x_2x_3-5x_4x_5=0,\\[2mm] & x_1^3+x_2^3+1=0.\end{cases}$$

该问题有最优解 $x^*=(-1.71,1.59,1.82,-0.763,-0.763)^T$, 最优值 $f(x^*)=0.0539$.

解 编制 Matlab 程序如下

```
function [x,mu,val,k]=lagsqp(x0,mu0)
%功能: 用基于拉格朗日函数Hesse阵的SQP方法求解约束优化问题:
%       min f(x)      s.t. h_i(x)=0, i=1,..., l.
%输入:  x0是初始点, mu0是乘子向量的初始值
%输出:  x, mu 分别是近似最优点及相应的乘子,
%  val是最优值, mh是约束函数的模, k是迭代次数.
maxk=50;   %最大迭代次数
n=length(x0); l=length(mu0);
rho=0.5; gamma=0.2;
x=x0; mu=mu0;  tau=0.1;
k=0;  epsilon=1e-8;
while(k<maxk)
    P1=P(x,mu)  %计算罚函数的值
    if(P1<epsilon), break; end  %检验终止准则
    H=B(x,mu,tau);    % 计算KT矩阵
    c=df1(x);     %计算目标函数梯度
    be=-h1(x); %计算约束函数
    Ae=dh1(x); %计算约束函数的Jacobi矩阵
    [dx,lam]=qsubp(H,c,Ae,be);
    du=lam-mu-1.0/(2*tau)*dh1(x)*dx;
    m=0; mk=0;
    while(m<20)   % Armijo搜索
        if(P(x+rho^m*dx,mu+rho^m*du)<=(1-gamma*rho^m)*P1)
            mk=m; break;
        end
        m=m+1;
    end
    x=x+rho^mk*dx; mu=mu+rho^mk*du;
    k=k+1;
end
val=f1(x);
%%%%%%%% 求解子问题 %%%%%%%%%%%%%%
function [x,mu1]=qsubp(H,c,Ae,be)
ginvH=pinv(H);
[m,n]=size(Ae);
if(m>0)
```

```
    rb=Ae*ginvH*c + be;
    mu1=pinv(Ae*ginvH*Ae')*rb;
    x=ginvH*(Ae'*mu1-c);
else
    x=-ginvH*c;
    mu1=zeros(m,1);
end
%%%%%%%%% 拉格朗日函数 L(x,mu) %%%%%%%%%%%%%
function l=la(x,mu)
f=f1(x);   %调用目标函数文件
h=h1(x);  %调用约束函数文件
l=f-mu'*h; % 计算乘子函数
%%%%%%%%% 拉格朗日函数的梯度 %%%%%%%%%%%%%
function dl=dla(x,mu)
df=df1(x); %调用目标函数梯度文件
h=h1(x);  %调用约束函数文件
dh=dh1(x);  %调用约束函数Jacobi矩阵文件
dl=[df-dh'*mu; -h];  %计算乘子函数梯度文件
%%%%%%%%% 罚函数P(x,mu) %%%%%%%%%%%%%%%
function s=P(x,mu)
dl=dla(x,mu);
s=norm(dl)^2;
%%%%%%%%% 拉格朗日函数的Hesse阵 %%%%%%%%%%%
function d2l=d2la(x,mu)
d2f=d2f1(x);   %调用目标函数Hesse阵文件
[d2h1,d2h2,d2h3]=d2h(x); %调用约束函数二阶导数文件
d2l=d2f-mu(1)*d2h1-mu(2)*d2h2-mu(3)*d2h3;
%%%%%%%%% KT矩阵B(x,mu) %%%%%%%%%%%
function H=B(x,mu,tau)  %计算KT矩阵
d2l=d2la(x,mu);   %计算Hesse阵
dh=dh1(x);  %约束函数的Jacobi矩阵
H=d2l+1.0/(2*tau)*dh'*dh;
%%%%%%%%% 目标函数 f(x) %%%%%%%%%%%%%
function f=f1(x)
s=x(1)*x(2)*x(3)*x(4)*x(5);
f=exp(s)-0.5*(x(1)^3+x(2)^3+1)^2;
%%%%%%%%% 约束函数 h(x) %%%%%%%%%%%%%
```

```
function h=h1(x)
h=[x(1)^2+x(2)^2+x(3)^2+x(4)^2+x(5)^2-10; ...
   x(2)*x(3)-5*x(4)*x(5);x(1)^3+x(2)^3+1];
%%%%%%%%% 目标函数 f(x) 的梯度%%%%%%%%%%%%%
function df=df1(x)
s=x(1)*x(2)*x(3)*x(4)*x(5);
df(1)=s/(x(1))*exp(s)-3*(x(1)^3+x(2)^3+1)*x(1)^2;
df(2)=s/(x(2))*exp(s)-3*(x(1)^3+x(2)^3+1)*x(2)^2;
df(3)=s/(x(3))*exp(s);
df(4)=s/(x(4))*exp(s);
df(5)=s/(x(5))*exp(s);
df=df(:);
%%%%%%%%% 约束函数 h(x) 的Jacobi矩阵A(x)%%%%%%%%%%%%
function dh=dh1(x)
dh=[2*x(1),2*x(2),2*x(3),2*x(4),2*x(5);0,x(3),x(2),-5*x(5),-5*x(4); ...
    3*x(1)^2,3*x(2)^2,0,0,0];
%%%%%%%%% 目标函数 f(x) 的Hesse阵%%%%%%%%%%%%%
function d2f=d2f1(x)
s=x(1)*x(2)*x(3)*x(4)*x(5);
d2f=[(s/(x(1)))^2*exp(s)-6*x(1)*(x(1)^3+x(2)^3+1)-9*x(1)^4,...
    (1+s)*x(3)*x(4)*x(5)*exp(s)-9*x(1)^2*x(2)^2, ...
    (1+s)*x(2)*x(4)*x(5)*exp(s),...
    (1+s)*x(2)*x(3)*x(5)*exp(s),(1+s)*x(2)*x(3)*x(4)*exp(s); ...
    (1+s)*x(3)*x(4)*x(5)*exp(s)-9*x(1)^2*x(2)^2,  ...
    (s/(x(2)))^2*exp(s)-6*x(2)*(x(1)^3+x(2)^3+1)-9*x(2)^4, ...
    (1+s)*x(1)*x(4)*x(5)*exp(s),(1+s)*x(1)*x(3)*x(5)*exp(s), ...
    (1+s)*x(1)*x(3)*x(4)*exp(s); ...
    (1+s)*x(2)*x(4)*x(5)*exp(s),(1+s)*x(1)*x(4)*x(5)*exp(s), ...
     s^2/(x(3))*exp(s),(1+s)*x(1)*x(2)*x(5)*exp(s), ...
    (1+s)*x(1)*x(2)*x(4)*exp(s); ...
    (1+s)*x(2)*x(3)*x(5)*exp(s),(1+s)*x(1)*x(3)*x(5)*exp(s), ...
    (1+s)*x(1)*x(2)*x(5)*exp(s),s^2/(x(4))*exp(s), ...
    (1+s)*x(1)*x(2)*x(3)*exp(s);...
    (1+s)*x(2)*x(3)*x(4)*exp(s),(1+s)*x(1)*x(3)*x(4)*exp(s), ...
    (1+s)*x(1)*x(2)*x(4)*exp(s),(1+s)*x(1)*x(2)*x(3)*exp(s), ...
     s^2/(x(5))*exp(s)]';
%%%%%%%%%%%%%%%约束函数h(x)的Hesse阵%%%%%%%%%%%%%%%%%%
```

```
function [d2h1,d2h2,d2h3]=d2h(x)
d2h1=[2 0 0 0 0; 0 2 0 0 0; 0 0 2 0 0; 0 0 0 2 0; 0 0 0 0 2]';
d2h2=[0 0 0 0 0; 0 0 1 0 0; 0 1 0 0 0; 0 0 0 0 -5; 0 0 0 -5 0]';
d2h3=[6*x(1) 0 0 0 0;0 6*x(2) 0 0 0;0 0 0 0 0;0 0 0 0 0;0 0 0 0 0]';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

利用上面的程序, 取乘子向量的初值为 $\mu_0=(0,0,0)$, 终止准则值取为 $P(x_k,\mu_k)\le 10^{-12}$, 对于不同的初始点得到计算结果如下表所示.

初始点 (x_0)	迭代次数 (k)	$f(x_k)$ 的值	$P(x_k,\mu_k)$ 的值
$(-1.71.51.8-0.6-0.6)^T$	11	0.0539	4.7981e-013
$(-1.7,1.6,1.8,-0.7,-0.7)^T$	11	0.0539	8.6424e-014
$(-1.8,1.7,1.9,-0.8,-0.8)^T$	9	0.0539	4.4646e-013
$(-2,1.5,2,-1,-1)^T$	14	0.0539	2.1538e-013
$(-3,2,3,-2,-2)^T$	16	0.0539	5.3309e-013

12.2.2　基于修正 Hesse 阵的 SQP 方法

首先, 考虑将上一节中关于构造二次规划子问题求解等式约束优化问题的思想推广到一般形式的约束优化问题 (12.1). 在给定点 (x_k,μ_k,λ_k) 之后, 将约束函数线性化, 并且对拉格朗日函数进行二次多项式近似, 得到下列形式的二次规划子问题:

$$\begin{aligned} \min\quad & \frac{1}{2}d^TW_kd+\nabla f(x_k)^Td \\ \text{s.t}\quad & h_i(x_k)+\nabla h_i(x_k)^Td=0,\ i\in E, \\ & g_i(x_k)+\nabla g_i(x_k)^Td\ge 0,\ i\in I, \end{aligned} \tag{12.16}$$

其中 $E=\{1,\cdots,l\}$, $I=\{1,\cdots,m\}$, $W_k=W(x_k,\mu_k,\lambda_k)=\nabla^2_{xx}L(x_k,\mu_k,\lambda_k)$, 而拉格朗日函数为

$$L(x,\mu,\lambda)=f(x)-\sum_{i\in E}\mu_ih_i(x)-\sum_{i\in I}\lambda_ig_i(x).$$

于是, 迭代点 x_k 的校正步 d_k 以及新的拉格朗日乘子估计量 μ_{k+1}, λ_{k+1} 可以分别定义为问题 (12.16) 的最优解 d^* 和相应的拉格朗日乘子 μ^*, λ^*.

上述的二次规划子问题 (12.16) 可能不存在可行点, 为了克服这一困难, Powell 引进了一个辅助变量 ξ, 然后求解下面的线性规划

$$\begin{aligned} \min \quad & -\xi \\ \text{s.t.} \quad & -\xi h_i(x_k) + \nabla h_i(x_k)^T d = 0, \ i \in E, \\ & -\xi g_k(x_k) + \nabla g_i(x_k)^T d \geq 0, \ i \in U_k, \\ & \ g_k(x_k) + \nabla g_i(x_k)^T d \geq 0, \ i \in V_k, \\ & \ -1 \leq \xi \leq 0, \end{aligned} \tag{12.17}$$

其中 $U_k = \{i | g_i(x_k) < 0, \ i \in I\}$, $V_k = \{i | g_i(x_k) \geq 0, \ i \in I\}$. 显然 $\xi = 0$, $d = 0$ 上述线性规划的一个可行点, 并且该线性规划的极小点 $\bar{\xi} = -1$ 当且仅当二次规划子问题 (12.16) 是相容的, 即子问题的可行域非空.

当 $\bar{\xi} = -1$ 时, 可以用线性规划问题的最优解 $\bar{d}$ 作为初始点, 求出二次规划子问题的最优解 d_k. 而当 $\bar{\xi} = 0$ 或接近于 0 时, 二次规划子问题无可行点, 此时需要重新选择迭代初始点 x_k, 然后启动 SQP 算法. 当 $\bar{\xi} \neq -1$ 但比较接近 -1 时, 可以用对应 $\bar{\xi}$ 的约束条件代替原来的约束条件, 再求解修正后的二次规划子问题.

不失一般性, 以后为了讨论的方便, 我们假设二次规划子问题是相容的. 下面的定理描述了迭代点对应的有效约束指标集与最优有效约束指标集之间的关系，其证明可参见文献 [15].

定理 62　给定约束优化问题 (12.1) 的一个 KT 点 x^* 和相应的拉格朗日乘子向量 μ^*, $\lambda^* \geq 0$. 假定在 x^* 处, 下面的条件成立:

(1) 有效约束的 Jacobi 矩阵 $J_{S(x^*)}$ 行满秩, 此处 $S(x^*) = E \cup I(x^*)$.

(2) 严格互补松弛条件成立, 即 $g_i(x^*) \geq 0$, $\lambda_i^* \geq 0$, $\lambda_i^* g_i(x^*) = 0$, $\lambda_i^* + g_i(x^*) > 0$.

(3) 二阶最优性充分条件成立, 即对任意满足 $A(x^*)d = 0$ 的向量 $d \neq 0$, 成立

$$d^T W(x^*, \mu^*, \mu^*) d > 0.$$

那么, 若 (x_k, μ_k, λ_k) 充分靠近 (x^*, μ^*, λ^*), 则二次规划子问题 (12.16) 存在一个局部极小点 d^*, 使得其对应的有效约束指标集 $S(d^*)$ 与原问题在 x^* 处的有效约束指标集 $S(x^*)$ 是相同的.

注意到在构造二次规划子问题 (12.16) 时, 需要计算拉格朗日函数在迭代点 x_k 处的 Hesse 阵 $W_k = W(x_k, \mu_k, \lambda_k)$, 其计算量是巨大的. 为了克服这一缺陷, 1976 年华裔数学家韩世平 (S.P. Han) 基于牛顿-拉格朗日方法提出了一种利用对称正定矩阵 B_k 替代拉格朗日矩阵 W_k 的序列二次规划法. 另外由于 R.B. Wilson 在 1963 年较早地考虑了牛顿-拉格朗日方法, 加之 M.J.D. Powell 于 1977 年修正了 Han 的方法, 所以人们也将这种序列二次规划法称之为 WHP 方法 (Wilson-Han-Powell 方法).

对于一般约束的优化问题 (12.1), 在迭代点 (x_k, μ_k, λ_k) 处, WHP 方法需要构造一个下列形式的二次规划子问题:

$$\begin{aligned} \min \quad & \frac{1}{2}d^T B_k d + \nabla f(x_k)^T d \\ \text{s.t.} \quad & h_i(x_k) + \nabla h_i(x_k)^T d = 0, \ i \in E, \\ & g_i(x_k) + \nabla g_i(x_k)^T d \geq 0, \ i \in I, \end{aligned} \tag{12.18}$$

并且用该二次规划子问题的解 d_k 作为原问题的变量 x 在第 k 次迭代过程中的搜索方向. 顺便提一下, 这个搜索方向 d_k 具有一个比较好的性质: 即它是许多罚函数 (价值函数) 的下降方向, 比如 ℓ_1 罚函数 (价值函数)

$$P_\sigma(x) = f(x) \ + \ \frac{1}{\sigma}\left[\sum_{i\in E}|h_i(x)| + \sum_{i\in I}\big|[g_i(x)]_-\big|\right], \tag{12.19}$$

其中罚参数 $\sigma > 0$, $[g_i(x)]_- = \max\{0, -g_i(x)\}$.

现在我们写出 WHP 方法的算法步骤如下.

算法 34 (WHP 方法)

步 0 给定初始点 $x_0 \in \mathbb{R}^n$, 初始对称矩阵 $B_0 \in \mathbb{R}^{n\times n}$, 容许误差限 $0 \leq \varepsilon \ll 1$ 和满足 $\sum_{k=0}^{\infty} \eta_k < +\infty$ 的非负数列 $\{\eta_k\}$. 取参数 $\sigma > 0$ 和 $\delta > 0$. 置 $k := 0$.

步 1 求解子问题 (12.18), 得最优解 d_k.

步 2 若 $\|d_k\| \leq \varepsilon$, 停算, 输出 x_k 作为近似极小点.

步 3 利用 ℓ_1 罚函数 $P_\sigma(x)$, 按照某种线搜索规则确定步长 $\alpha_k \in (0, \delta]$, 使得

$$P_\sigma(x_k + \alpha_k d_k) \leq \min_{\alpha\in(0,\delta]} P_\sigma(x_k + \alpha d_k) + \eta_k.$$

步 4 置 $x_{k+1} := x_k + \alpha_k d_k$, 更新 B_k 为 B_{k+1}.

步 5 令 $k := k + 1$, 转步 1.

在一定的条件下, 可以证明算法 34 的全局收敛性. 下面, 我们不加证明地给出算法 34 的全局收敛性定理, 其证明过程可参见文献 [1].

定理 63　对于约束优化问题 (12.1), 假设 $f, h_i\,(i\in E)$ 和 $g_i\,(i\in I)$ 都是连续可微的, 且存在常数 $0<m\le M$, 使得算法 34 中的对称正定矩阵 B_k 满足

$$m\|d\|^2 \le d^T B_k d \le M\|d\|^2,\ \ (\forall\, d\in\mathbb{R}^n,\ k=1,2,\cdots).$$

若罚参数 $\sigma>0$ 和二次规划子问题 (12.18) 的拉格朗日乘子向量 $\mu_{k+1},\ \lambda_{k+1}\ge 0$ 满足

$$\sigma\max\{\|\lambda_{k+1}\|_\infty, \|\mu_{k+1}\|_\infty\}\le 1,\ \ (\forall\, k=1,2,\cdots).$$

则算法 34 生成的序列 $\{x_k\}$ 的任何聚点都是问题 (12.1) 的 KT 点.

12.3　SQP 方法的相关问题

12.3.1　二次规划子问题的 Hesse 阵

下面以纯等式的约束优化问题为例, 说明如何选择二次规划子问题的 Hesse 阵, 其中二次规划子问题为

$$\begin{aligned} \min\quad & \frac{1}{2}d^T W_k d + \nabla f(x_k)^T d \\ \text{s.t}\quad & h_i(x_k) + \nabla h_i(x_k)^T d = 0,\ i=\{1,\cdots,l\}. \end{aligned} \tag{12.20}$$

.

1. 基于拟牛顿校正公式的选择方法

由于拟牛顿法是处理无约束优化问题的有效算法, 故可利用拟牛顿法的基本原理, 对拉格朗日函数 $L(x_k,\mu_k)$ 的 Hesse 阵 $\nabla^2_{xx}L(x_k,\mu_k)$ 的近似矩阵 B_k 进行修正. 令

$$s_k = x_{k+1}-x_k,\ \ y_k = \nabla_x L(x_{k+1},\mu_{k+1}) - \nabla_x L(x_k,\mu_{k+1}). \tag{12.21}$$

因 BFGS 校正公式要求向量 s_k 和 y_k 满足曲率条件, 即 $s_k^T y_k>0$, 但由 (12.21) 确定的向量 s_k, y_k 可能不满足这一条件. 为此, 有必要对向量 y_k 进行修正. 下面的修正策略是由 Powell 于 1978 年给出的: 令

$$z_k = \theta_k y_k + (1-\theta_k)B_k s_k, \tag{12.22}$$

其中参数 θ_k 定义为

$$\theta_k = \begin{cases} 1, & 若\ s_k^T y_k \geq 0.2 s_k^T B_k s_k, \\ \dfrac{0.8 s_k^T B_k s_k}{s_k^T B_k s_k - s_k^T y_k}, & 若\ s_k^T y_k < 0.2 s_k^T B_k s_k. \end{cases} \tag{12.23}$$

于是, 矩阵 B_k 的约束 BFGS 校正公式为

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{z_k z_k^T}{s_k^T z_k}. \tag{12.24}$$

注意到参数 θ_k 的定义 (12.23), 不难验证: 当 $\theta = 1$时,

$$s_k^T z_k = s_k^T y_k \geq 0.2 s_k^T B_k s_k > 0,$$

而当 $\theta_k \neq 1$ 时,

$$s_k^T z_k = \theta_k s_k^T y_k + (1 - \theta_k) s_k^T B_k s_k = 0.2 s_k^T B_k s_k > 0.$$

因此, 约束 BFGS 校正公式 (12.24) 可以保持正定性.

2. 基于增广拉格朗日函数的选择方法

下面考虑增广拉格朗日函数

$$L_A(x, \mu, \sigma) = f(x) - \mu^T h(x) + \frac{1}{2\sigma}\|h(x)\|^2, \tag{12.25}$$

其中罚参数 $\sigma > 0$, $h(x) = (h_1(x), \cdots, h_l(x))^T$. 在 KT 点 (x^*, μ^*) 处, 根据 $h(x^*) = 0$, 可知增广拉格朗日函数的 Hesse 阵为

$$\nabla_{xx}^2 L_A(x^*, \mu^*, \sigma) = \nabla_{xx}^2 L(x^*, \mu^*) + \frac{1}{\sigma} A(x^*)^T A(x^*). \tag{12.26}$$

对于纯等式约束的优化问题, 若约束函数在 x^* 处的 Jacobi 矩阵 $A(x^*)$ 行满秩, 并且二阶最优性二阶充分性条件成立, 则存在某个阈值 $\bar{\sigma} > 0$, 使得 $\forall \sigma \in (0, \bar{\sigma}]$, $\nabla_{xx}^2 L_A(x_k, \mu_k, \sigma)$ 是正定的. 于是二次规划子问题中的 W_k 可取为 $\nabla_{xx}^2 L_A(x_k, \mu_k, \sigma)$, 或者取对 $\nabla_{xx}^2 L_A(x_k, \mu_k, \sigma)$ 进行拟牛顿近似的校正矩阵 B_k.

12.3.2　价值函数与搜索方向的下降性

为了保证序列二次规划法 (SQP 方法) 的全局收敛性, 通常借助于某价值函数来确定搜索步长. 例如, 目标函数、罚函数以及增广拉格朗日函数等都可以作为价值函数, 用来衡量一维搜索的好坏. 下面介绍两类著名的价值函数.

1. ℓ_1 价值函数

首先, 以纯等式约束的优化问题为例, 考虑下列 ℓ_1 价值函数

$$\phi_1(x) = f(x) + \frac{1}{\sigma}\|h(x)\|_1, \tag{12.27}$$

其中 $\sigma > 0$ 为罚参数.

命题 2　设 d_k 和 μ_{k+1} 分别是子问题 (12.20) 的最优解和拉格朗日乘子, 则 ϕ_1 沿方向 d_k 的方向导数满足

$$\phi_1'(x_k, \sigma; d_k) \le -d_k^T W_k d_k - (\sigma^{-1} - \|\mu_{k+1}\|_\infty)\|h(x_k)\|_1. \tag{12.28}$$

证　由泰勒公式可得

$$\begin{aligned} f(x_k + \alpha d_k) &= f(x_k) + \nabla f(x_k)^T d_k + \frac{1}{2}\alpha^2 d_k^T \nabla^2 f(\xi_k) d_k, \\ h(x_k + \alpha d_k) &= h(x_k) + \alpha A(x_k) d_k + \frac{1}{2}\alpha^2 d_k^T \nabla^2 h(\eta_k) d_k \\ &= (1-\alpha)h(x_k) + \frac{1}{2}\alpha^2 d_k^T \nabla^2 h(\eta_k) d_k, \end{aligned}$$

其中

$$d_k^T \nabla^2 h(\eta_k) d_k = \left[d_k^T \nabla^2 h_1(\eta_k) d_k,\ d_k^T \nabla^2 h_2(\eta_k) d_k,\ \cdots,\ d_k^T \nabla^2 h_l(\eta_k) d_k\right]^T.$$

故对比较小的 $\alpha > 0$, 有

$$\phi_1(x_k + \alpha d_k) - \phi_1(x_k) \le \alpha(\nabla f(x_k)^T d_k - \sigma^{-1}\|h(x_k)\|_1) + \alpha^2 \vartheta \|d_k\|^2,$$

其中 $\vartheta > 0$ 是某个常数. 同理可证

$$\phi_1(x_k + \alpha d_k) - \phi_1(x_k) \ge \alpha(\nabla f(x_k)^T d_k - \sigma^{-1}\|h(x_k)\|_1) - \alpha^2 \vartheta \|d_k\|^2.$$

根据多元函数方向导数的定义, 可得

$$\phi_1'(x_k, \sigma; d_k) = \nabla f(x_k)^T d_k - \sigma^{-1}\|h(x_k)\|_1. \tag{12.29}$$

由 KT 条件, 有

$$\nabla f(x_k)+W_kd_k-A(x_k)^T\mu_{k+1}=0,$$

于是有

$$\begin{aligned}\nabla f(x_k)^Td_k &= -d_k^TW_kd_k+d_k^TA(x_k)^T\mu_{k+1}\\ &= -d_k^TW_kd_k-h(x_k)^T\mu_{k+1}.\end{aligned}$$

注意到由 Schwartz 不等式, 有

$$-h(x_k)^T\mu_{k+1}\le \|h(x_k)\|_\infty\|\mu_{k+1}\|_\infty\le \|\mu_{k+1}\|_\infty\|h(x_k)\|_1.$$

至此, 命题的结论已得到证明. □

根据命题 2 可知, 若矩阵 W_k 正定, 且取罚参数

$$\sigma^{-1}\ge \|\mu_{k+1}\|_\infty+\bar\delta,$$

其中 $\bar\delta>0$, 则可保证 d_k 是 $\phi_1(x,\sigma)$ 在相应点 x_k 处的下降方向. 此外, 在构造二次规划子问题 (12.20) 时, 如果用对称正定矩阵 B_k 替代 W_k, 那么相应的最优解 d_k 也可以是价值函数 ϕ_1 的下降方向.

对于一般的约束优化问题 (12.1), 可以考虑相应的二次规划子问题 (12.18), 并且将 (12.19) 定义的罚函数作为价值函数:

$$\begin{aligned}P_\sigma(x) &= f(x)+\sigma^{-1}\big[\|h(x)\|_1+\|g(x)_-\|_1\big]\\ &= f(x)\ +\ \frac{1}{\sigma}\left[\sum_{i\in E}|h_i(x)|+\sum_{i\in I}\big|[g_i(x)]_-\big|\right]\end{aligned}$$

其中 $h(x)=(h_1(x),\cdots,h_l(x))^T$, $g(x)=(g_1(x),\cdots,g_m(x))^T$, $g(x)_-=\max\{0,-g_i(x)\}$.

命题 3 设 d_k 和 μ_{k+1}, $\lambda_{k+1}\ge 0$ 分别是子问题 (12.18) 的最优解和拉格朗日乘子, 则 P_σ 沿 d_k 的方向导数满足

$$\begin{aligned}P_\sigma'(x_k;d_k) \le\ & -d_k^TB_kd_k-(\sigma^{-1}-\|\mu_{k+1}\|_\infty)\|h(x_k)\|_1\\ & -(\sigma^{-1}-\|\lambda_{k+1}\|_\infty)\|g(x_k)_-\|_1.\end{aligned}\tag{12.30}$$

证 注意到 1-范数 $\|\cdot\|_1$ 是凸函数以及函数 $(\cdot)_-$ 具有下列性质

$$\forall\, t_1\ge t_2,\ \text{有}\ \ (t_1)_-\le (t_2)_-.$$

利用与命题 2 类似的方法可得

$$P'_\sigma(x_k;d_k)=\nabla f(x_k)^Td_k-\sigma^{-1}\big[\|h(x_k)\|_1+\|g(x_k)_-\|_1\big]. \tag{12.31}$$

再利用二次规划子问题 (12.18) 的 KT 条件, 有

$$\begin{cases}\nabla f(x_k)+B_kd_k=\nabla h(x_k)^T\mu_{k+1}+\nabla g(x_k)^T\lambda_{k+1},\\ \lambda_{k+1}\ge 0,\ \ g(x_k)+\nabla g(x_k)^Td_k\ge 0,\\ \lambda_{k+1}^T[g(x_k)+\nabla g(x_k)^Td_k]=0.\end{cases} \tag{12.32}$$

又因 d_k 满足可行性条件: $h(x_k)+\nabla h(x_k)^Td_k=0$, 由此及 (12.32) 的第一、三两式可得

$$\nabla f(x_k)^Td_k=-d_k^TB_kd_k-h(x_k)^T\mu_{k+1}-g(x_k)^T\lambda_{k+1}. \tag{12.33}$$

由 Schwartz 不等式可知

$$\begin{cases}-h(x_k)^T\mu_{k+1}=\|h(x_k)\|_\infty\|\mu_{k+1}\|_\infty\le\|\mu_{k+1}\|_\infty\|h(x_k)\|_1,\\ -g(x_k)^T\lambda_{k+1}\le\lambda_{k+1}^T[g(x_k)]_-\le\|\lambda_{k+1}\|_\infty\|g(x_k)\|_1.\end{cases} \tag{12.34}$$

最后, 由 (12.31), (12.33) 和 (12.34) 即得命题的结论. □

2. Fletcher 价值函数

Fletcher 价值函数也叫做增广拉格朗日价值函数. SQP 方法问世后, 除了使用 ℓ_1 价值函数以得到算法的全局收敛性外, 优化工作者还引进过其他的价值函数. 例如, Fletcher 曾经针对纯等式约束优化问题引入了下列增广拉格朗日价值函数:

$$\phi_F(x,\sigma)=f(x)-\mu(x)^Th(x)+\frac{1}{2\sigma}\|h(x)\|^2, \tag{12.35}$$

其中 $\mu(x)$ 是乘子向量, $\sigma>0$ 是罚参数. 若函数 $h(x)$ 的 Jacobi 矩阵 $A(x)=\nabla h(x)^T$ 是行满秩的, 则乘子向量可取为

$$\mu(x)=[A(x)A(x)^T]^{-1}A(x)\nabla f(x), \tag{12.36}$$

即 $\mu(x)$ 是下面的最小二乘问题的解:

$$\min_{\mu\in\mathbb{R}^l}\|\nabla f(x)-A(x)^T\mu\|.$$

另一方面, Fletcher 函数在 x_k 处的梯度为

$$\nabla \phi_F(x_k, \sigma) = \nabla f(x_k) - A_k^T \mu(x_k) - \nabla \mu(x_k) h(x_k) + \sigma^{-1} A_k^T h(x_k), \tag{12.37}$$

其中 $A_k = A(x_k)$ 是 $h(x)$ 在 x_k 处的 Jacobi 矩阵, 且 A_k 行满秩. 假设 d_k 是子问题

$$\begin{aligned} \min \quad & \frac{1}{2} d^T B_k d + \nabla f(x_k)^T d \\ \text{s.t.} \quad & h(x_k) + A_k d = 0 \end{aligned} \tag{12.38}$$

的最优解, 并记

$$d_k = A_k^T d_k^y + Z_k d_k^z,$$

其中 Z_k 的列向量是零空间 $\mathcal{N}(A_k)$ 的一组基. 根据子问题 (12.38) 的可行性条件, 我们有

$$d_k^y = -(A_k A_k^T)^{-1} h(x_k).$$

然后, 利用最小二乘乘子 $\mu(x_k)$ 的定义 (12.36) 可得

$$\nabla f(x_k)^T A_k^T d_k^y = -\mu(x_k)^T h(x_k).$$

故有

$$\nabla \phi_F(x_k, \sigma)^T d_k = \nabla f(x_k)^T Z_k d_k^z - h(x_k)^T \nabla \mu(x_k)^T d_k - \sigma^{-1} \|h(x_k)\|^2. \tag{12.39}$$

另由 KT 条件有

$$W_k d_k + \nabla f(x_k) - A_k \mu_k = 0,$$

其中 μ_k 是拉格朗日乘子. 那么, 我们有

$$\begin{aligned} Z_k^T W_k d_k &= Z_k^T W_k Z_k d_k^z + Z_k^T W_k A_k^T d_k^y \\ &= Z_k^T [A_k \mu_k - \nabla f(x_k)] \\ &= -Z_k^T \nabla f(x_k). \end{aligned}$$

于是, 有

$$\begin{aligned} \nabla \phi_F(x_k, \sigma)^T d_k &= -(d_k^z)^T [Z_k^T W_k Z_k] d_k^z - (d_k^y)^T A_k W_k Z_k d_k^z \\ &\quad - h(x_k)^T \nabla \mu(x_k)^T d_k - \sigma^{-1} \|h(x_k)\|^2. \end{aligned} \tag{12.40}$$

因此, 当简约 Hesse 阵 $Z_k^T W_k Z_k$ 正定时, 若取罚参数满足

$$\sigma^{-1} \geq \left[\frac{-(d_k^y)^T A_k W_k Z_k d_k^z - h(x_k)^T \nabla\mu(x_k)^T d_k}{\|h(x_k)\|^2}\right] + \bar{\delta}, \tag{12.41}$$

其中 $\bar{\delta} > 0$, 则 d_k 是 Fletcher 价值函数的下降方向.

下面我们给出一般形式约束优化问题的 SQP 方法的算法步骤.

算法 35 (一般约束优化问题的 SQP 方法)

步 0　给定初始点 $(x_0, \mu_0, \lambda_0) \in \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m$, 对称正定矩阵 $B_0 \in \mathbb{R}^{n\times n}$. 计算

$$A_0^E = \nabla h(x_0)^T,\ \ A_0^I = \nabla g(x_0)^T,\ \ A_0 = \begin{bmatrix} A_0^E \\ A_0^I \end{bmatrix}.$$

选择参数 $\eta \in (0, 1/2)$, $\rho \in (0,1)$, 容许误差 $0 \leq \varepsilon_1, \varepsilon_2 \ll 1$. 令 $k := 0$.

步 1　求解子问题

$$\begin{aligned} \min\ \ & \frac{1}{2} d^T B_k d + \nabla f(x_k)^T d \\ \text{s.t.}\ \ & h(x_k) + A_k^E d = 0, \\ & g(x_k) + A_k^I d \geq 0, \end{aligned} \tag{12.42}$$

得最优解 d_k.

步 2　若 $\|d_k\|_1 \leq \varepsilon_1$ 且 $\|h_k\|_1 + \|(g_k)_-\|_1 \leq \varepsilon_2$, 停算, 得到原问题的的一个近似 KT 点 (x_k, μ_k, λ_k).

步 3　对于某种价值函数 $\phi(x, \sigma)$, 选择罚参数 σ_k, 使得 d_k 是该函数在 x_k 处的下降方向.

步 4　Armijo 搜索. 令 m_k 是使下列不等式成立的最小非负整数 m:

$$\phi(x_k + \rho^m d_k, \sigma_k) - \phi(x_k, \sigma_k) \leq \eta\rho^m \phi'(x_k, \sigma; d_k), \tag{12.43}$$

令 $\alpha_k := \rho^{m_k}$, $x_{k+1} := x_k + \alpha_k d_k$.

步 5　计算

$$A_{k+1}^E = \nabla h(x_{k+1})^T,\ \ A_{k+1}^I = \nabla g(x_{k+1})^T,\ \ A_{k+1} = \begin{bmatrix} A_{k+1}^E \\ A_{k+1}^I \end{bmatrix}.$$

以及最小二乘乘子

$$\begin{bmatrix} \mu_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \left[A_{k+1}A_{k+1}^T\right]^{-1} A_{k+1}\nabla f_{k+1}. \tag{12.44}$$

步 6 校正矩阵 B_k 为 B_{k+1}. 令

$$s_k = \alpha_k d_k, \quad y_k = \nabla_x L(x_{k+1}, \mu_{k+1}, \lambda_{k+1}) - \nabla_x L(x_k, \mu_{k+1}, \lambda_{k+1}),$$

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{z_k z_k^T}{s_k^T z_k}, \tag{12.45}$$

其中

$$z_k = \theta_k y_k + (1-\theta_k) B_k s_k, \tag{12.46}$$

参数 θ_k 定义为

$$\theta_k = \begin{cases} 1, & \text{若 } s_k^T y_k \geq 0.2 s_k^T B_k s_k, \\ \dfrac{0.8 s_k^T B_k s_k}{s_k^T B_k s_k - s_k^T y_k}, & \text{若 } s_k^T y_k < 0.2 s_k^T B_k s_k. \end{cases} \tag{12.47}$$

步 7 令 $k := k+1$, 转步 1.

注 (1) 算法35 步 5 隐含地假设了 A_k 是行满秩的. 如果这个条件不成立, 则在计算最小二乘乘子时, 需要使用计算广义你的技巧, 即此时 (12.44) 为

$$\begin{bmatrix} \mu_{k+1} \\ \lambda_{k+1} \end{bmatrix} = (A_{k+1}^+)^T \nabla f_{k+1}, \tag{12.48}$$

其中 A_{k+1}^+ 是 A_{k+1} 的 Penrose-Moore 广义逆.

(2) 算法35 步 3 弱选择 ℓ_1 价值函数

$$\phi(x,\sigma) = f(x) + \sigma^{-1}\left[\|h(x)\|_1 + \|g(x)_-\|_1\right],$$

可令

$$\tau = \max\{\|\mu_k\|, \|\lambda_k\|\},$$

任意选择一个 $\delta > 0$, 定义罚参数的修正规则为

$$\sigma_k = \begin{cases} \sigma_{k-1}, & 若\ \sigma_{k-1}^{-1} \geq \tau + \delta, \\ (\tau + 2\delta)^{-1}, & 若\ \sigma_{k-1}^{-1} < \tau + \delta. \end{cases} \tag{12.49}$$

在本节最后, 需要指出的是, 对于无约束优化问题, 所谓的"超线性收敛步"成立, 即如果 x^* 是 $f(x)$ 的稳定点, 且 Hesse 阵 $\nabla^2 f(x^*)$ 正定, 那么只要迭代序列 $\{x_k\} \to x^*$, 且搜索方向满足

$$\lim_{k\to\infty} \frac{|x_k + d_k - x^*\|}{\|x_k - x^*\|} = 0,$$

对于充分大的 k, 就必然成立

$$f(x_k + d_k) < f(x_k).$$

换言之, 对于无约束优化问题, 超线性收敛步总是可以接受的. 但是对于约束优化问题情况并非如此, 即对于有些约束优化问题, 不管 x_k 如何靠近 x^*, 都不会有

$$\phi(x_k + d_k, \sigma) \leq \phi(x_k, \sigma)$$

成立, 且 $x_k + d_k$ 对应的目标函数值和可行度比 x_k 对应的目标函数值和可行度还要差一些. 也就是说, 对于约束优化问题, 在这种情况下, 超线性收敛步是无法接受的, SQP 方法会失去收敛阶高的优点, 人们把这种现象称之为 Maratos 效应.

为了克服 Maratos 效应, 人们已经提出了许多方法, 如放松接受试探步 $x_k + d_k$ 的条件 (Watchdog 技术); 或者引进满足 $\|\tilde{d}_k\| = O(\|d_k\|^2)$ 的二阶校正步 $\tilde{d}_k$, 使得相应的价值函数值 $\phi(x_k + d_k + \tilde{d}_k, \sigma) < \phi(x_k, \sigma)$; 或者使用光滑的精确罚函数作为价值函数, 以提高超线性收敛步接受度, 例如, Schittkowski 提出增广拉格朗日函数

$$\begin{aligned} \phi(x, v, r) \;\; = \;\; & f(x) - \sum_{j\in E}\Big(v_j h_j(x) - \frac{1}{2} r_j h_j^2(x)\Big) \\ & - \sum_{j\in J(x,v)} \Big(v_j g_j(x) - \frac{1}{2} r_j g_j^2(x)\Big) - \frac{1}{2}\sum_{j\in K(x,v)} v_j^2 / r_j, \end{aligned} \tag{12.50}$$

其中

$$J(x,v) = \{j \in I \,|\, g_j(x) \leq v_j/r_j\}, \quad K(x,v) = \{j \in I \,|\, g_j(x) > v_j/r_j\}.$$

一般来说, ℓ_1 精确罚函数形式简单, 但是非光滑, 增广拉格朗日函数是光滑的, 数值计算结果较好.

12.4 SQP 方法的 Matlab 程序

本节讨论 SQP 方法的 Matlab 实现. 注意到算法 35 每一迭代步的主要计算量是求解子问题 (12.42), 因此, 我们先讨论子问题的 Matlab 实现.

12.4.1 SQP 子问题的 Matlab 实现

利用 KT 条件, 问题 (12.42) 等价于

$$H_1(d,\mu,\lambda) = B_k d - (A_k^E)^T\mu - (A_k^I)^T\lambda + \nabla f(x_k) = 0, \tag{12.51}$$

$$H_2(d,\mu,\lambda) = h(x_k) + A_k^E d = 0, \tag{12.52}$$

$$\lambda \geq 0,\ g(x_k) + A_k^I d \geq 0,\ \ \lambda^T[g(x_k) + A_k^I d] = 0. \tag{12.53}$$

注意到 (12.53) 是一个 m 维的线性互补问题, 我们定义光滑 FB-函数

$$\phi(\varepsilon,a,b) = a + b - \sqrt{a^2+b^2+2\varepsilon^2},$$

其中 $\varepsilon > 0$ 是光滑参数. 令

$$\Phi(\varepsilon,d,\lambda) = (\phi_1(\varepsilon,d,\lambda), \phi_2(\varepsilon,d,\lambda),\cdots,\phi_m(\varepsilon,d,\lambda))^T,$$

其中

$$\phi_i(\varepsilon,d,\lambda) = \lambda_i + [g_i(x_k) + (A_k^I)_i d] - \sqrt{\lambda_i^2 + [g_i(x_k) + (A_k^I)_i d]^2 + 2\varepsilon^2},$$

其中 $(A_k^I)_i$ 表示矩阵 A_k^I 的第 i 行. 记 $z = (\varepsilon,d,\mu,\lambda) \in \mathbb{R}_+ \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l$. 那么方程组 (12.51)-(12.52) 等价于

$$H(z) := H(\varepsilon,d,\mu,\lambda) = \begin{bmatrix} \varepsilon \\ H_1(d,\mu,\lambda) \\ H_2(d,\mu,\lambda) \\ \Phi(\varepsilon,d,\lambda) \end{bmatrix} = 0. \tag{12.54}$$

不难计算出 $H(z)$ 的 Jacobi 矩阵为

$$H'(z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & B_k & -(A_k^E)^T & -(A_k^I)^T \\ 0 & A_k^E & 0 & 0 \\ v & D_2(z)A_k^I & 0 & D_1(z) \end{bmatrix}, \tag{12.55}$$

其中 $v=\nabla_\varepsilon\Phi(\varepsilon,d,\lambda)=(v_1,\cdots,v_m)^T$, v_i 由下式确定

$$v_i=-\frac{2\varepsilon}{\sqrt{\lambda_i^2+[g_i(x_k)+(A_k^I)_i d]^2+2\varepsilon^2}}, \tag{12.56}$$

而 $D_1(z)=\mathrm{diag}(a_1(z),\cdots,a_m(z))$, $D_2(z)=\mathrm{diag}(b_1(z),\cdots,b_m(z))$, 其中 $a_i(z),b_i(z)$ 由下式确定

$$a_i(z)=1-\frac{\lambda_i}{\sqrt{\lambda_i^2+[g_i(x_k)+(A_k^I)_i d]^2+2\varepsilon^2}}, \tag{12.57}$$

$$b_i(z)=1-\frac{g_i(x_k)+(A_k^I)_i d}{\sqrt{\lambda_i^2+[g_i(x_k)+(A_k^I)_i d]^2+2\varepsilon^2}}. \tag{12.58}$$

给定参数 $\gamma\in(0,1)$, 定义非负函数

$$\beta(z)=\gamma\|H(z)\|\min\{1,\|H(z)\|\}. \tag{12.59}$$

算法 36 (**求解子问题的光滑牛顿法**)

步 0 选取 $\rho,\ \eta\in(0,1)$, $\varepsilon_0>0$, $(d_0,\mu_0,\lambda_0)\in\mathbb{R}^n\times\mathbb{R}^l\times\mathbb{R}^m$. 置 $z_0=(\varepsilon_0,d_0,\mu_0,\lambda_0)$, $\bar z=(\varepsilon_0,0,0,0)$. 选取 $\gamma\in(0,1)$ 使 $\gamma\mu_0<1$ 及 $\gamma\|H(z_0)\|<1$. 令 $j:=0$.

步 1 如果 $\|H(z_j)\|=0$, 算法终止; 否则, 计算 $\beta_j=\beta(z_j)$.

步 2 求解下列方程组得解 $\Delta z_j=(\Delta\varepsilon_j,\Delta d_j,\Delta\mu_j,\Delta\lambda_j,)$,

$$H(z_j)+H'(z_j)\Delta z_j=\beta_j\bar z. \tag{12.60}$$

步 3 设 m_j 为满足下式的最小非负整数:

$$\|H(z_j+\rho^{m_j}\Delta z_j)\|\le[1-\sigma(1-\beta\mu_0)\rho^{m_j}]\|H(z_j)\|. \tag{12.61}$$

令 $\alpha_j:=\rho^{m_j}$, $z_{j+1}=z_j+\alpha_j\Delta z_j$.

步 4 令 $j:=j+1$, 转步 1.

下面我们给出算法 36 的 Matlab 程序.

程序 18 利用光滑牛顿法求解二次规划子问题.

```
function  [d,mu,lam,val,k]=qpsubp(dfk,Bk,Ae,hk,Ai,gk)
% 功能: 求解二次规划子问题:  min qk(d)=0.5*d'*Bk*d+dfk'*d+,
%          s.t.   hk+Ae*d=0,  gk+Ai*d>=0.
%输入: dfk是xk处的梯度, Bk是第k次近似Hesse阵, Ae,hk线性等式约束
%          的有关参数, Ai,gk是线性不等式约束的有关参数
%输出: d,val分别是是最优解和最优值, mu,lam是乘子向量, k是迭代次数.
n=length(dfk);  l=length(hk); m=length(gk);
gamma=0.05;  epsilon=1.0e-6;  rho=0.5;  sigma=0.2;
ep0=0.05;  mu0=0.05*zeros(l,1); lam0=0.05*zeros(m,1);
d0=ones(n,1);  u0=[ep0;zeros(n+l+m,1)];
z0=[ep0; d0; mu0;lam0,];
k=0;  %k为迭代次数
z=z0; ep=ep0; d=d0; mu=mu0;  lam=lam0;
while (k<=150)
   dh=dah(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk);
   if(norm(dh)<epsilon)
       break;
   end
   A=JacobiH(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk);
   b=beta(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk,gamma)*u0-dh;
   dz=A\b;
   if(l>0&m>0)
       de=dz(1); dd=dz(2:n+1); du=dz(n+2:n+l+1); dl=dz(n+l+2:n+l+m+1);
   end
   if(l==0)
      de=dz(1);   dd=dz(2:n+1);  dl=dz(n+2:n+m+1);
   end
   if(m==0)
      de=dz(1);   dd=dz(2:n+1);  du=dz(n+2:n+l+1);
   end
   i=0; %mk=0;
   while (mm<=20)
     if(l>0&m>0)
      dh1=dah(ep+rho^i*de,d+rho^i*dd,mu+rho^i*du,lam+rho^i*dl,dfk,Bk,Ae,hk,Ai,gk);
     end
     if(l==0)
       dh1=dah(ep+rho^i*de,d+rho^i*dd,mu,lam+rho^i*dl,dfk,Bk,Ae,hk,Ai,gk);
```

```
        end
        if(m==0)
            dh1=dah(ep+rho^i*de,d+rho^i*dd,mu+rho^i*du,lam,dfk,Bk,Ae,hk,Ai,gk);
        end
        if(norm(dh1)<=(1-sigma*(1-gamma*ep0)*rho^i)*norm(dh))
            mk=i; break;
        end
        i=i+1;
        if(i==20), mk=10; end
    end
    alpha=rho^mk;
    if(l>0&m>0)
        ep=ep+alpha*de;    d=d+alpha*dd;
        mu=mu+alpha*du;   lam=lam+alpha*dl;
    end
    if(l==0)
        ep=ep+alpha*de;    d=d+alpha*dd;
        lam=lam+alpha*dl;
    end
    if(m==0)
        ep=ep+alpha*de;    d=d+alpha*dd;
        mu=mu+alpha*du;
    end
    k=k+1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%
function p=phi(ep,a,b)
p=a+b-sqrt(a^2+b^2+2*ep^2);
%%%%%%%%%%%%%%%%%%%%%%%%%%
function dh=dah(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk)
n=length(dfk); l=length(hk); m=length(gk);
dh=zeros(n+l+m+1,1);
dh(1)=ep;
if(l>0&m>0)
   dh(2:n+1)=Bk*d-Ae'*mu-Ai'*lam+dfk;
   dh(n+2:n+l+1)=hk+Ae*d;
   for(i=1:m)
```

```
      dh(n+l+1+i)=phi(ep,lam(i),gk(i)+Ai(i,:)*d);
   end
end
if(l==0)
   dh(2:n+1)=Bk*d-Ai'*lam+dfk;
   for(i=1:m)
     dh(n+1+i)=phi(ep,lam(i),gk(i)+Ai(i,:)*d);
   end
end
if(m==0)
   dh(2:n+1)=Bk*d-Ae'*mu+dfk;
   dh(n+2:n+l+1)=hk+Ae*d;
end
dh=dh(:);
%%%%%%%%%%%%%%%%%%%%%%%%%%
function bet=beta(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk,gamma)
dh=dah(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk);
bet=gamma*norm(dh)*min(1,norm(dh));
%%%%%%%%%%%%%%%%%%%%%%%%%%
function [dd1,dd2,v1]=ddv(ep,d,lam,Ai,gk)
m=length(gk);
dd1=zeros(m,m); dd2=zeros(m,m); v1=zeros(m,1);
for(i=1:m)
    fm=sqrt(lam(i)^2+(gk(i)+Ai(i,:)*d)^2+2*ep^2);
    dd1(i,i)=1-lam(i)/fm;
    dd2(i,i)=1-(gk(i)+Ai(i,:)*d)/fm;
    v1(i)=-2*ep/fm;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%
function A=JacobiH(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk)
n=length(dfk); l=length(hk); m=length(gk);
A=zeros(n+l+m+1,n+l+m+1);
[dd1,dd2,v1]=ddv(ep,d,lam,Ai,gk);
if(l>0&m>0)
    A=[1,            zeros(1,n),  zeros(1,l),  zeros(1,m);
       zeros(n,1),   Bk,          -Ae',        -Ai';
       zeros(l,1),   Ae,         zeros(l,l),  zeros(l,m) ;
```

```
            v1,            dd2*Ai,       zeros(m,l),  dd1];
    end
    if(l==0)
         A=[1,            zeros(1,n),  zeros(1,m);
            zeros(n,1), Bk,           -Ai';
            v1,          dd2*Ai,       dd1];
    end
    if(m==0)
        A=[1,             zeros(1,n), zeros(1,l);
           zeros(n,1),   Bk,           -Ae';
           zeros(l,1),   Ae,         zeros(l,l)];
    end
```

下面我们利用程序 18, 求解三个分别是纯等式约束、纯不等式约束以及混合约束的二次规划问题.

例 38　解二次规划问题

$$\begin{aligned} \min \quad & f(x) = x_1^2 + 2x_2^2 + x_3^2 - 2x_1x_2 + x_3 \\ \text{s.t.} \quad & x_1 + x_2 + x_3 - 4 = 0, \\ & 2x_1 - x_2 + x_3 - 2 = 0. \end{aligned}$$

该问题的极小点为 $x^* = \left(\dfrac{21}{11}, \dfrac{43}{22}, \dfrac{3}{22}\right)^T$.

解　在 Matlab 命令窗口依次输入下列命令:

```
dfk=[0 0 1]';
Bk=[2 -2 0; -2 4 0; 0 0 2];
Ae=[1 1 1; 2 -1 1];
hk=[-4 -2]';
Ai=[];
gk=[];
[d,mu,lam,val,k]=qpsubp(dfk,Bk,Ae,hk,Ai,gk)
```

得到计算结果:

```
d =
   1.9091
   1.9545
```

```
    0.1364
mu =
    2.6364
   -1.3636
lam =
    Empty matrix: 0-by-1
val =
    3.9773
k =
    2
```

例 39 解二次规划问题

$$
\begin{aligned}
\min\quad & f(x)=\frac{1}{2}x_1^2-x_1x_2+x_2^2-6x_1-2x_2\\
\text{s.t.}\quad & -2x_1-x_2+3\ge 0,\\
& x_1-x_2+1\ge 0;\\
& -x_1-2x_2+2\ge 0;\\
& x_1,x_2\ge 0.
\end{aligned}
$$

该问题的极小点为 $x^*=\left(\frac{4}{3},\frac{1}{3}\right)^T$.

解 在 Matlab 命令窗口依次输入下列命令:

```
dfk=[-6 -2]';
Bk=[1 -1; -1 2];
Ae=[ ];
hk=[ ]';
Ai=[-2 -1;1 -1;-1 -2; 1 0; 0 1];
gk=[3 1 2 0 0]';
[d,mu,lam,val,k]=qpsubp(dfk,Bk,Ae,hk,Ai,gk)
```

得到计算结果:

```
d =
   1.3333
   0.3333
mu =
```

```
  Empty matrix: 0-by-1
lam =
  2.4444
  0.0000
  0.1111
  0.0000
  0.0000
val =
  -8.1111
k =
  5
```

例 40 解二次规划问题

$$\begin{aligned} \min \quad & f(x) = x_1^2 + x_1x_2 + 2x_2^2 - 6x_1 - 2x_2 - 12x_3 \\ \text{s.t.} \quad & x_1 + x_2 + x_3 - 2 = 0, \\ & x_1 - 2x_2 + 3 \geq 0; \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

该问题的极小点为 $x^* = (0,0,2)^T$.

解 在 Matlab 命令窗口依次输入下列命令:

```
dfk=[-6 -2 -12]';
Bk=[2 1 0;1 4 0; 0 0 0];
Ae=[1 1 1];
hk=[-2]';
Ai=[1 -2 0;1 0 0; 0 1 0;0 0 1];
gk=[3 0 0 0]';
[d,mu,lam,val,k]=qpsubp(dfk,Bk,Ae,hk,Ai,gk)
```

得到计算结果:

```
d =
   0.0000
   0.0000
   2.0000
mu =
```

```
    12.0000
lam =
    0.0000
    6.0000
    10.0000
    0.0000
val =
  -24.0000
k =
  6
```

12.4.2 SQP 方法的 Matlab 实现

本小节给出 SQP 方法 (算法 35) 的一个 Matlab 程序, 该程序在某种意义上是通用的, 不同的问题只需编写目标函数、约束函数以及它们的梯度和 Jacobi 矩阵的 M 文件即可调用该程序.

程序 19 一般约束优化问题 SQP 方法的 Matlab 程序, 该程序在每一迭代步调用了程序 18 qpsubp.m 求解二次规划子问题.

```
function [x,mu,lam,val,k]=sqpm(x0,mu0,lam0)
%功能: 用基于拉格朗日函数Hesse阵的SQP方法求解约束优化问题:
%       min f(x)      s.t. h_i(x)=0, i=1,..., l.
%输入: x0是初始点, mu0是乘子向量的初始值
%输出: x, mu分别是近似最优点及相应的乘子,
%val是最优值, mh是约束函数的模, k是迭代次数.
maxk=100;    %最大迭代次数
n=length(x0); l=length(mu0); m=length(lam0);
rho=0.5; eta=0.1;  B0=eye(n);
x=x0; mu=mu0;  lam=lam0;
Bk=B0; sigma=0.8;
epsilon1=1e-6; epsilon2=1e-5;
[hk,gk]=cons(x);  dfk=df1(x);
[Ae,Ai]=dcons(x); Ak=[Ae; Ai];
k=0;
while(k<maxk)
   [dk,mu,lam]=qpsubp(dfk,Bk,Ae,hk,Ai,gk);  %求解子问题
   mp1=norm(hk,1)+norm(max(-gk,0),1);
```

```
if(norm(dk,1)<epsilon1)&(mp1<epsilon2)
    break;
end  %检验终止准则
deta=0.05;  %罚参数更新
tau=max(norm(mu,inf),norm(lam,inf));
if(sigma*(tau+deta)<1)
    sigma=sigma;
else
    sigma=1.0/(tau+2*deta);
end
im=0;   %Armijo搜索
while(im<=20)
  if(phi1(x+rho^im*dk,sigma)-phi1(x,sigma)<eta*rho^im*dphi1(x,sigma,dk))
    mk=im;
    break;
  end
  im=im+1;
  if(im==20),   mk=10;   end
end
alpha=rho^mk; x1=x+alpha*dk;
[hk,gk]=cons(x1);  dfk=df1(x1);
[Ae,Ai]=dcons(x1);  Ak=[Ae; Ai];
lamu=pinv(Ak)'*dfk;  %计算最小二乘乘子
if(l>0&m>0)
   mu=lamu(1:l); lam=lamu(l+1:l+m);
end
if(l==0), mu=[]; lam=lamu;  end
if(m==0), mu=lamu; lam=[];  end
sk=alpha*dk;  %更新矩阵Bk
yk=dlax(x1,mu,lam)-dlax(x,mu,lam);
if(sk'*yk>0.2*sk'*Bk*sk)
   theta=1;
else
   theta=0.8*sk'*Bk*sk/(sk'*Bk*sk-sk'*yk);
end
zk=theta*yk+(1-theta)*Bk*sk;
Bk=Bk+zk*zk'/(sk'*zk)-(Bk*sk)*(Bk*sk)'/(sk'*Bk*sk);
```

```
   x=x1;  k=k+1;
end
val=f1(x);
%p=phi1(x,sigma)
%dd=norm(dk)
%%%%%%%% l1精确价值函数 %%%%%%
function p=phi1(x,sigma)
f=f1(x); [h,g]=cons(x); gn=max(-g,0);
l0=length(h);  m0=length(g);
if(l0==0), p=f+1.0/sigma*norm(gn,1); end
if(m0==0),  p=f+1.0/sigma*norm(h,1); end
if(l0>0&m0>0)
   p=f+1.0/sigma*(norm(h,1)+norm(gn,1));
end
%%%%% 价值函数的方向导数%%%%%
function dp=dphi1(x,sigma,d)
df=df1(x); [h,g]=cons(x);  gn=max(-g,0);
l0=length(h);  m0=length(g);
if(l0==0),  dp=df'*d-1.0/sigma*norm(gn,1); end
if(m0==0), dp=df'*d-1.0/sigma*norm(h,1); end
if(l0>0&m0>0)
    dp=df'*d-1.0/sigma*(norm(h,1)+norm(gn,1));
end
%%%%%%%%% 拉格朗日函数 L(x,mu) %%%%%%%%%%%%%
function l=la(x,mu,lam)
f=f1(x);   %调用目标函数文件
[h,g]=cons(x);  %调用约束函数文件
l0=lemgth(h);  m0=length(g);
if(l0==0), l=f-lam*g;  end
if(m0==0),  l=f-mu'*h;  end
if(l0>0&m0>0)
   l=f-mu'*h-lam'*g;
end
%%%%%%%%% 拉格朗日函数的梯度 %%%%%%%%%%%%
function dl=dlax(x,mu,lam)
df=df1(x); %调用目标函数梯度文件
[Ae,Ai]=dcons(x);  %调用约束函数Jacobi矩阵文件
```

```
[m1,m2]=size(Ai); [l1,l2]=size(Ae);
if(l1==0), dl=df-Ai'*lam; end
if(m1==0), dl=df-Ae'*mu; end
if(l1>0&m1>0), dl=df-Ae'*mu-Ai'*lam; end
```

下面我们利用程序 19 来计算两个约束优化问题的极小点.

例 41 解非线性规划问题

$$\begin{aligned} \min \quad & f(x) = -\pi x_1^2 x_2 \\ \text{s.t.} \quad & \pi x_1 x_2 + \pi x_1^2 - 150 = 0, \\ & x_1 \geq 0,\ x_2 \geq 0. \end{aligned}$$

解 首先编写四个 m 函数:

```
%%%%%%%%目标函数f(x)%%%%%%%%%%%
function f=f1(x)  %f1.m
f=-pi*x(1)^2*x(2);
%%%%%%%目标函数f(x)的梯度%%%%%%%%
function df=df1(x)  %df1.m
df=[-2*pi*x(1)*x(2),-pi*x(1)^2]';
%%%%%%%%%约束函数 %%%%%%%%%%%%%%
function [h,g]=cons(x)  %cons.m
h=[pi*x(1)*x(2)+pi*x(1)^2-150];
g=[x(1);x(2)];
%%%%%%%%约束函数Jacobi矩阵%%%%%%%%
function [dh,dg]=dcons(x) %dcons.m
dh=[pi*x(2)+2*pi*x(1), pi*x(1)];
dg=[1 0; 0 1];
```

在 Matlab 命令窗口依次输入下列命令:

```
x0=[3 3]';
mu0=[0]';
lam0=[0 0]';
[x,mu,lam,val,k]=sqpm(x0,mu0,lam0)
```

得到计算结果:

```
x =
   3.9894
   7.9788
mu =
   -3.9894
lam =
   1.0e-007 *
   0.4248
   0.0039
val =
   -398.9423
k =
   8
```

例 42 解非线性规划问题

$$
\begin{aligned}
\min \quad & f(x) = x_1^2 + x_2^2 - 16x_1 - 10x_2 \\
\text{s.t.} \quad & -x_1^2 + 6x_1 - 4x_2 + 11 \geq 0, \\
& x_1x_2 - 3x_2 - \mathrm{e}^{x_1-3} + 1 \geq 0, \\
& x_1 \geq 0,\ x_2 \geq 0.
\end{aligned}
$$

解 首先编写四个 m 函数:

```
%%%%%%%%目标函数f(x)%%%%%%%%%%
function f=f1(x)  %f1.m
f=x(1)^2+x(2)^2-16*x(1)-10*x(2);
%%%%%%%目标函数f(x)的梯度%%%%%%%%
function df=df1(x)  %df1.m
df=[2*x(1)-16; 2*x(2)-10];
%%%%%%%%%约束函数 %%%%%%%%%%%%%%
function [h,g]=cons(x)  %cons.m
h=[ ];
g=[-x(1)^2+6*x(1)-4*x(2)+11; ...
   x(1)*x(2)-3*x(2)-exp(x(1)-3)+1;x(1);x(2)];
%%%%%%%约束函数Jacobi矩阵%%%%%%%%
function [dh,dg]=dcons(x) %dcons.m
dh=[ ];
dg=[-2*x(1)+6,-4;x(2)-exp(x(1)-3),x(1)-3;1,0;0,1];
```

在 Matlab 命令窗口依次输入下列命令:

```
x0=[4 4]';
mu0=[ ]';
lam0=[0 0 0 0]';
[x,mu,lam,val,k]=sqpm(x0,mu0,lam0)
```

得到计算结果:

```
x =
   5.2396
   3.7460
mu =
   Empty matrix: 0-by-1
lam =
   0.8133
   0.3327
   0.0000
   0.0000
val =
   -79.8078
k =
   5
```

习 题 12

1. 用 SQP 方法求解问题

$$\begin{aligned} \min \quad & f(x) = x_1 + x_2, \\ \text{s.t.} \quad & x_2 - x_1^2 \geq 0. \end{aligned}$$

初始点为 $x_0 = (0,0)^T$. B_k 用两种取法

(1) 取 $B_k = \nabla_{xx}^2 L(x_k, \lambda_k)$, 且 $\lambda_0 = 1$; 若取 $\lambda_0 = 0$, 结果怎样?

(2) 取 $B_0 = I$ (单位阵), 并采用校正公式 (12.45) $\sim$ (12.47) 校正 B_k.

2. 用线搜索全局 SQP 方法求解问题

$$\begin{aligned} \min \quad & f(x) = -x_1 - x_2, \\ \text{s.t.} \quad & -x_1^2 + x_2 \geq 0, \\ & -x_1^2 - x_2^2 + 1 \geq 0. \end{aligned}$$

此问题的最优解为 $x^*=\left(\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2}\right)^T$. 取初始点 $x_0=\left(\frac{1}{2},1\right)^T$, $\lambda_0=(0,0)^T$, B_k 用两种方式计算:

(1) $B_k=\nabla_{xx}^2 L(x_k,\lambda_k)$.

(2) 取$B_0=I$ (单位阵), 并采用校正公式 (12.45) ∼ (12.47) 校正 B_k.

3. 考虑最优化问题

$$\begin{aligned} \min\quad & f(x)=-x_1+2(x_1^2+x_2^2-1),\\ \text{s.t.}\quad & x_1^2+x_2^2-1=0. \end{aligned}$$

若取价值函数为 Fletcher 精确罚函数, 试问单位步长能否使价值函数下降?

4. 设等式约束优化问题为

$$\begin{aligned} \min\quad & f(x)=x_1x_2^2,\\ \text{s.t.}\quad & g(x)=x_1^2+x_2^2-2=0. \end{aligned}$$

(1) 取初始点 $x_0=(-2,-2)^T$. 初始矩阵 $B_0=I$ (单位阵). 用步长为 1 的 SQP 方法求解此问题.

(2) 取 $\bar{x}=(-0.8,-1,1)^T$, $\lambda=-0.8$. 求 $r^*>0$, 使得 $0<r<r^*$ 时,

$$\nabla_{xx}^2 L(\bar{x},\lambda)+\frac{1}{r}\nabla g(\bar{x})\nabla g(\bar{x})^T$$

正定.

5. 用光滑牛顿法的 Matlab 程序求解二次规划子问题

$$\begin{aligned} (1)\quad \min\quad & f(x)=\frac{1}{2}(x_1^2+2x_2^2-2x_1x_2+x_3^2),\\ \text{s.t.}\quad & x_1+x_2-x_3=4,\\ & x_1-2x_2+x_3=-2. \end{aligned}$$

$$\begin{aligned} (2)\quad \min\quad & f(x)=x_1^2+x_2^2,\\ \text{s.t.}\quad & -2x_1-x_2+2\geq 0,\\ & x_2-1\geq 0 \end{aligned}$$

6. 用 SQP 方法的 Matlab 程序求解下列优化问题

$$\begin{aligned} (1)\quad \min\quad & f(x)=(x_1-2)^4+(x_1-2x_2)^2,\\ & \text{s.t.}\ -x_1^2+x_2\geq 0. \end{aligned}$$

取初始点 $x_0=(0,0)^T$, 初始矩阵 $B_0=I$ (单位阵).

$$\begin{aligned}(2)\quad \min\quad & f(x)=e^{-x_1-x_2}+x_1^2+2x_1x_2+x_2^2+2x_1+6x_2,\\ \text{s.t.}\quad & 2-x_1-x_2\ge 0,\\ & x_1,x_2\ge 0.\end{aligned}$$

取初始点 $x_0=(2,0)^T$, 初始矩阵 $B_0=I$ (单位阵).

附录 A　Matlab 优化工具箱简介

在科学工程领域中, 优化问题有着十分广泛的应用. 根据数学理论定义, 优化是指在某种约束条件下, 寻求目标函数的最大值或者最小值. 将上面的定义转换为数学公式:

$$\min_{x\in S} f(x),$$

在上面的公式中, x 表示变量向量, 即 $x=(x_1,\cdots,x_n)$, $f(x)$ 是优化情况下的目标函数, S 表示的是所承受的约束条件. 如果没有接受任何的约束条件, 或者 S 是数值条件下的全集, 则该优化为无约束优化; 否则, 则是约束优化. 本节首先介绍无约束优化的内容, 然后约束优化的内容, 最后, 将介绍线性规划和二次规划这两种在实际中比较常见的优化问题.

本附录将简要介绍 Matlab 软件的 Optimization Toolbox (优化工具箱) 的使用方法, 所涉及的内容将都是 MATLAB 内置的函数, 有些比较复杂的优化处理工具会涉及 Optimization Toolbox 中的函数和内容. 如果希望自行演示本附录中的程序代码, 请选择安装 Optimization Toolbox 组件.

A.1　线性规划

线性规划是一种特殊的优化问题, 目标函数和约束条件都是线性的. 对于这种优化问题, 可以使用比较特使的方法来求解, 典型的线性规划问题为:

$$\min_{x}\ f(x)=c^Tx$$
$$\text{s.t.}\begin{cases} A\cdot x\le b,\\ Ae\cdot x=be,\\ lb\le x\le ub.\end{cases}$$

在 MATLAB 优化工具箱中, 求解线性规划的命令为 linprog, 其完整的调用格式如下:

[x,fval,exitflag,output,lambda]=linprog(c,A,b,Ae,be,lb,ub,x0,options)

输入参数: 参数 c 表示目标函数中的常向量, x0 表示的是优化的初始值, 参数 A, b 表示的是满足线性关系式 $Ax \le b$ 的系数矩阵合结果矩阵; 参数 Ae, be 表示的式满足线性等式 $Ae \cdot x = be$ 的矩阵; 参数 lb, ub 则表示满足参数取值范围 $lb \le x \le ub$ 的上限和下限; 参数 options 就是进行优化的属性设置;

输出参数: exitflag 表示程序退出优化运算的类型, output 参数包含多种关于优化的信息, 包含 iterations 等; 参数 lambda 则表示各种约束问题的拉各朗日参数数值.

例 43 求解线性规划, 其中目标函数为 $f(x) = -3x_1 - 2x_2$, 其中参数满足下面的关系式: $0 \le x_1,\ x_2 \le 10$. 同时, 该目标函数满足下面的约束条件:

$$\begin{cases} 3x_1 + 4x_2 \le 7, \\ 2x_1 + x_2 \le 3, \\ -3x_1 + 2x_2 = 2. \end{cases}$$

1. 在 MATLAB 的命令窗口中输入下面的程序代码:

```
x0=[0 0]';
c=[-3 -2]';
A=[3 4;2 1];
b=[7 3]';
Ae=[-3 2];
be=2;
lb=[0 0]';
ub=[10 10]';
[x,fval,exitflag,output,lambda]=linprog(c,A,b,Ae,be,lb,ub,x0);
```

2. 查看线性规划求解的结果. 在命令窗口中输入下面的程序代码

x, fval

说明: 从上面的结果中可以看出, 使用线性规划求解的最优解为 (0.3333, 1.500), 对应的最优解对应的函数值为 -4.

说明: 函数 linprog 的输入参数 x0 是可选的, 我们看下面的例子.

例 44 求解下面的现行线性规划:

$$\begin{cases} \min & -5x_1+4x_2+2x_3, \\ \text{s.t.} & 6x_1-x_2+x_3\le 8, \\ & x_1+2x_2+4x_3\le 10, \\ & -1\le x_1\le 3, \\ & 0\le x_2\le 2, \\ & 0\le x_3. \end{cases}$$

在 MATLAB 的命令窗口中输入下面的程序代码:

```
c=[-5 4 2]';
A=[6 -1 1;1 2 4];  b=[8 10]';
Ae=[ ]; be=[ ];
lb=[-1 0 0]';
ub=[3 2]';
[x,fval,exitflag,output,lambda]=linprog(c,A,b,Ae,be,lb,ub);
```

A.2　二次规划

二次规划是指其目标函数 2 次多项式而约束函数全部是线性函数的优化问题, 标准的二次规划格式如下:

$$\begin{aligned} &\min_x f(x)=\frac{1}{2}x^T Hx+c^T x \\ &\text{s.t. } A\cdot x\le b \\ &\quad Ae\cdot x=be \\ &\quad lb\le x\le ub \end{aligned}$$

在 MATLAB 中, 求解二次规划的命令为 quadprog, 其完整的调用格式如下:
[x,fval,exitflag,output,lambda]=quadprog(c,A,b,Aeq,beq,lb,ub,x0,options)
关于该命令中的各个参数的含义, 请参考前面的 linprog 命令.

例 45 求解二次规划, 其目标函数为:

$$\min f(x)=\frac{1}{2}x_1^2+x_2^2-x_1x_2-2x_1-6x_2$$

约束条件为:

$$\begin{cases} x_1 + x_2 \le 2, \\ -x_1 + 2x_2 \le 2, \\ 2x_1 + x_2 \le 3, \\ x_1 \ge 0,\ x_2 \ge 0. \end{cases}$$

1. 将二次规划进行转换, 转换为标准形式. 根据线性代数知识, 得到的结果为:

$$H = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}, \quad c = \begin{pmatrix} -2 \\ -6 \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

2. 进行二次规划求解. 在 MATLAB 的命令窗口中输入下面的代码:

```
H=[1 -1;-1 2];
c=[-2;-6];
A=[1 1;-1 2;2 1];
b=[2;2;3];
lb=[0;0];
[x,fval,exitflag,output,lambda]=quadprog(H,c,A,b,[],[],lb);
```

3. 查看二次规划求解的结果. 在 MATLAB 的命令窗口中输入代码:

x, fval

在上面的二次规划问题中, 求得的最优解为 (0.6667, 1.3333), 对应的函数数值为 −8.222.

A.3　无约束非线性优化

在上面介绍过, 无约束优化相当于约束集为全集. MATLAB 为解决非约束优化提供 fminsearch 和 fminunc 函数, 其对应的详细调用格式如下:

[x,fval,exitflag,output] = fminsearch(fun,x0,options)

在上面的命令格式中, 参数比较繁多, 下面分别分详细介绍.

输入参数: 参数“fun”表示优化的目标函数, 参数“”表示执行优化的初始数值, 参数“options”表示进行优化的各种属性, 一般需要使用 optimset 函数进行设置.

输出参数: 参数“x”表示最优解; “fval”表示最优解对应的函数值; 参数“exitflag”则表示函数退出优化运算的原因, 取值为 1、0 和 −1. 其中数值 1 表示函数收敛于最优解, 0 则表示函数迭代次数超过了优化属性的设置, −1 则表示优化迭代算法被 output 函数终止; 参数“output”是一种结构变量, 显示的是关于优化的属性信息, 例如优化迭代次数和优化算法等.

说明: 在 MATLAB 中, fminsearch 一般适用于没有约束条件的非线性优化情况, 对于线性优化的情况, 将在后面介绍.

fminunc 函数的调用格式如下:

[x,fval,exitflag,outflag,grad,hessian] = fminunc(fun,x0,options)

该函数的大部分参数的含义和 fminsearch 函数相同, 而输出参数“grad”表示的是函数在最优解下的梯度; 参数“hessian”则表示目标函数在最优解的 Hessian 矩阵数值; 参数“exitflag”表示的也是停止最优解的类型, 但是其取值包括了 −2, −1, 0, 1, 2 和 3. 因此, 该函数比上面的函数能够更加详细地描述优化情况. 关于其具体的含义, 可自行查看相应的帮助文件.

下面看一个无约束非线性优化实例

例 46 *求解二元函数 $f(x) = 3x_1^2 + 2x_1x_2 + x_2^2$ 在全集范围之内的最小值, 分别使用不同的优化函数和优化属性. 为了让读者能够直观查看优化求解情况, 可以在求解之后绘制二元函数的图形.*

1. 选择命令窗口编辑栏中的“File-New” - “M-File”命令, 打开 M 文件编辑器, 在其中输入下面的程序代码:

```
function[f,g]=optfun(x)
f=3*x(1)^2+2*x(1)*x(2)+x(2)^2;
if nargout>1
   g(1)=6*x(1)+2*x(2);
   g(2)=2*x(1)+2*x(2);
end
```

在上面的程序代码中, g 表示的是 f 函数的梯度.

在输入完上面的程序代码后, 将该代码保存为“optfun.m”文件.

2. 选择优化的初始数值 [1, 1], 分别使用不同的函数求解优化. 在 MATLAB 命令窗口中输入下面的程序代码:

x0=[1,1];

options=optimset('Display','iter','TolFun',1e-18,'GradObj','on');

[x,fval,exitflag,output,grad]=fminunc(@optfun,x0,options)

[x1,fval1,exitflag1,output1]=fminsearch(@optfun,x0,options)

说明: 在上面的结果中, 使用 fminuc 函数的最优解为 $(0,0)$, 而且迭代的次数为 2, 对应的优化求解方法为“large-scale:trust-region Newton”; 使用 fminsearch 函数求解的最优解为 $(0,0)$, 且迭代次数为 81, 使用的优化求解方法为“Nelder-Mead simplex direct search”. 因此, 在相同的初值条件下, 两个方法求解的性质不同.

3. 选择优化的初始值 $[-1,1]$, 分别使用不同的函数求解优化. 在 MATLAB 的命令窗口中输入下面的程序代码:

x0=[-1,1];

options=optimset('Display','iter','TolFun',1e-18,'GradObj','on');

[x,fval,exitflag,output,grad]=fminunc(@optfun,x0,options)

[x1,fval1,exitflag1,output1]=fminsearch(@optfun,x0,options)

从上面的程序结果中可以看出, 当修改优化的初始条件后, 各种优化函数使用的迭代次数会明显的改变. 因此, 设置初值将直接影响优化求解的效率.

A.4　非线性最小二乘问题

非线性最小二乘问题 (nonlinear least-squares) 是非线性约束优化的一种特例, 其优化的目标函数为

$$\min \sum_{i=1}^{n} f_i^2(x).$$

在 MATLAB 中, 为了求解非线性最小二乘问题, 提供 lsqnonlin 函数命令, 其最完整的调用格式如下:

[x,resnorm,residual,exitflag,output,lambada]=lsqnonlin(fun,x0,lb,ub,options)

在上面的命令中, 函数的参数比较复杂, 下面详细介绍各参数的具体含义.

输入参数: 参数 fun 表示优化的目标函数, 参数 x0 表示优化的初值条件, lb 是进行优化求解的下限, ub 是进行优化求解的上限, 相当于 lb≤x≤ub; 参数 options 则表示优化求解的优化属性.

输入参数: 参数 x 表示所求的最优解; 参数 resnorm 则表示二阶范数, 在数值上等于sum(fun(x).^{2}); 参数 residual 则表示优化求解后的残数; 参数 lambda 则表示在最优解处的拉各朗日数值; 其他参数和其他优化命令中含义相同.

提示: 在上面的命令格式中, 并不是所有的参数都是必须输入的. 在参数输入部分, 只有 fun, x0 是必须输入的, 而在输出参数部分, 则只有 x 是必须的. MATLAB 之所以提供上面的许多参数, 是为了方便用户自行研究和比较优化的属性.

下面看一个非线性最小二乘问题实例.

例 47 以函数 $f(x)=\dfrac{1}{1+8x^2}$ 为基准函数, 对其进行非线性最小二乘运算, 同时以最小二乘为目标进行数据拟合.

1. 选择命令窗口编辑栏中得“File-New” - “M-File”命令, 打开 M 文件编辑器, 在其中输入下面得程序代码:

```
function F=flq(a)
xx=-2+[0:200]/50;
F=polyval(a,xx)-1./(1+8*xx.*xx);
```

在输入上面的程序代码后, 将上面的程序代码保存为“flq.m”文件.

2. 进行非线性最小方差求解. 在 MATLAB 的命令窗口中输入下面的程序代码:

```
N=5;
a0=zeros(1,N);
lb=-1*ones(1,N);
ub= ones(1,N);
options=optimset('Display','iter','TolFun',1e-18,'Gradobj','on');
[x,resnorm,residual,exitflag,output,lambda]=lsqnonlin('flq',a0,lb,ub,options);
```

3. 查看程序结果. 在输入上面的程序代码后, 按“Eenter”键, 得到的结果如下:

```
x
resnorm
exitflag
output
```

A.5　约束条件的非线性优化命令

有约束条件的优化情况比无约束条件的优化情况要复杂得多, 处理起来也更困难, 种类也比较繁杂. 这里仅限于讨论 MATLAB 内置函数 fmincon 的使用方

法. 首先, fmincom 函数主要用于解决具有下面约束条件的优化

$$\min\ f(x)$$
$$\text{s.t.}\begin{cases} c(x)\le 0,\\ ce(x)=0,\\ A\cdot x\le b,\\ Ae\cdot x=be,\\ lb\le x\le ub.\end{cases}$$

该函数的完整调用格式如下:

[x,fval,exitflag,output,lambda]

=fmincon(fun,x0,A,b,Ae,be,lb,ub,nonlcon,options)

该函数的参数比较复杂, 下面详细介绍各种参数的含义.

输入参数: 参数 fun 表示的是优化目标函数, x0 表示的是优化的初始值, 参数 A, b 表示的是满足线性关系式 $Ax\le b$ 的系数矩阵合结果矩阵; 参数 Ae, be 表示的式满足线性等式 $Ae\cdot x=be$ 的矩阵; 参数 lb, ub 则表示满足参数取值范围 $lb\le x\le ub$ 的上限和下限; 参数 nonlcon 则表示需要参数满足的非线性关系式 $c(x)\le 0$ 和 $ce(x)=0$ 的优化情况; 参数 options 就是进行优化的属性设置;

输入参数: exitflag 表示程序退出优化运算的类型, 取值为 -2, -1, 0, 1, 2, 3, 4 和 5, 其数值对应的类型在此不作详细说明. output 参数包含多种关于优化的信息, 包含 iterations、funcCount、algorithm、cgiterations、stepsize 和 firstorderopt 等; 参数 lambda 则表示 lower、upper、ublneqlin、eqlin、ineqnonlin 和 eqnonlin等, 分别表示优化问题的各种约束问题的拉各朗日参数数值.

下面给出一个约束条件的非线性优化实例.

例 48 求解在约束条件 $0\le x_1+2x_2+3x_3\le 72$, 函数 $f(x)=-x_1x_2x_3$ 的最小值的最优解以及最优解的数值.

1. 转换约束条件, 将上面的约束条件转换为下面的关系式

$$\begin{cases} -x_1-2x_2-3x_3\le 0,\\ x_1+2x_2+3x_3\le 72.\end{cases}$$

提示: 由于在 fmincon 中, 所有的约束条件都是以上面的不等式形式出现的, 因此在本步骤中需要将原来的约束条件转换为不等式方程组.

2. 选择命令窗口编辑栏中的“File” - “New” - “M-File” 命令, 打开 M 文件安编辑器, 在其中输入下面的程序代码;

```
function f=optcon(x)
f=-x(1)*x(2)*x(3);
```

将上面的程序代码保存为“optcon.m”文件, 该文件将是最优解的目标函数.

3. 在 MATLAB 的命令窗口中输入下面的程序代码:

```
A=[-1 -2 -3; 1 2 3];
b=[0;72];
x0=[10;10;10];
[x,fval,exitflag,output,lambda]=fmincon(@optcon,x0,A,b);
```

4. 查看优化信息. 在输入上面的程序代码后, 按“Enter”键, MATLAB 将会进行优化运算, 并显示对应的优化信息;

在上面的程序中显示了实质起作用的约束条件和优化中止的类型.

5. 查看优化的结果. 在 MATLAB 的命令窗口中输入下面的程序代码;

```
x
fval
output
```

6. 重新设置优化条件, 进行优化运算. 将最优问题的约束条件修改为下面的关系式:

$$\begin{cases} x_2 \le 5, \\ x_3 \le 10, \\ x_1 + 2x_2 + 3x_3 \le 72. \end{cases}$$

同时, 将初值设置为 $[1, 1, 1]$, 然后进行优化求解, 得到的结果如下:

```
x0=[1,1,1];
A=[1,2,3;0,1,0;0,0,1];
b=[72;5;10];
[x,fval,exitflag,output,lambda]=fmincon(@optcon,x0,A,b);
x
fval
exitflag
output
```

提示: 从上面的条件可以看出, 当修改关于优化的各种属性后, 优化问题会发生质的改变, 因此在进行优化求解问题的时候, 需要特别注意优化求解的条件.

A.6　最小最大值的优化问题

最小最大值的优化问题函是以个比较特殊的问题, 其表示的是从一系列最大值中选取最小的数值, 相当于求解下面的优化问题:

$$\min_x \max_i \{F_i(x)\},$$

约束条件为:

$$\begin{cases} c(x) \leq 0, \\ ce(x) = 0, \\ A \cdot x \leq b, \\ Ae \cdot x = be, \\ lb \leq x \leq ub. \end{cases}$$

在上面的目标函数中, $F(x) = (F_1(x), F_2(x), \cdots, F_m(x))^T$. MATLAB 提供了函数 fminmax 来求解最小最大值优化问题, 该函数的参数和使用方法和前面介绍的 fmincon 完全相同, 这里不重复介绍. 下面使用简单的实例来介绍如何使用该函数命令.

例 49 *求解函数* $F(x) = (F_1(x), \cdots, F_5(x))$ *的最小最大值, 其中各分量函数依次为* $F_1(x) = 2x_1^2 + x_2^2 - 48x_1 - 40x_2 + 125$, $F_2(x) = -x_1^2 - 3x_2^2$, $F_3(x) = x_1 + 3x_2 - 18$, $F_4(x) = -x_1 - x_2$ *和* $F_5(x) = x_1 + x_2 - 8$.

1. 选择命令窗口中的“File” - “New” - “M-File” 命令, 打开 M 文件编辑器, 在其中输入下面的程序代码:

```
function f=mnmax(x)
f(1)=2*x(1)^2+x(2)^2-48*x(1)-40*x(2)+125;
f(2)=-x(1)^2-3*x(2)^2;
f(3)=x(1)+3*x(2)-18;
f(4)=-x(1)-x(2);
f(5)=x(1)+x(2)-8;
```

在输入上面的程序代码后, 将该代码保存为“mnmax.m”文件.

2. 求解最小最大值的优化问题, 在 MATLAB 的命令窗口中输入下面的程序代码:

```
x0=[0.1;0.1];
[x,fval]=fminimax(@mnmax,x0)
```

在上面的求解过程中, 首先设置了初值条件, 然后直接调用函数求解优化问题.

参考文献

[1] 袁亚湘, 孙文瑜. 最优化理论与方法. 北京: 科学出版社, 1997.

[2] 倪勤. 最优化方法与程序设计. 北京: 科学出版社, 2009.

[3] 徐成贤, 陈志平, 李乃成. 近代优化方法. 北京: 科学出版社, 2002.

[4] 李董辉, 童小娇, 万中. 数值最优化. 北京: 科学出版社, 2005.

[5] 赖炎连, 贺国平.最优化方法. 北京: 清华大学出版社, 2004.

[6] 孙文瑜, 徐成贤, 朱德通. 最优化方法. 北京: 高等教育出版社, 2004.

[7] 袁亚湘. 非线性优化计算方法. 北京: 科学出版社, 2008.

[8] 曹卫华, 郭正. 最优化技术方法及MATLAB的实现. 北京: 化学工业出版社, 2005.

[9] 陈宝林. 最优化理论与算法（第 2 版）. 北京: 清华大学出版社, 2006.

[10] 何勇坚. 最优化方法. 北京: 清华大学出版社, 2007.

[11] 唐焕文, 秦学志. 最优化方法. 大连: 大连理工大学出版社, 1994.

[12] 黄红选, 韩继业. 数学规划. 北京: 清华大学出版社, 2006.

[13] 张光澄, 王文娟, 韩会磊, 张雷. 非线性最优化计算方法. 北京: 高等教育出版社, 2005.

[14] John H. Mathews, Kurtis D. Fink 著, 周璐, 陈渝, 钱方等译. 数值方法 (MATLAB 版) (第四版). 北京: 电子工业出版社, 2007.

[15] Nocedal J, Wright S T. Numerical Optimization. Berlin: Springer-Verlag, 2000.